Markov and mixed models with applications

Mortensen, Stig Bousgaard

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Markov and mixed models with applications

Stig Bousgaard Mortensen

Kongens Lyngby 2009
IMM-PHD-2009-220

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-3192

Preface

This thesis was submitted at the Technical University of Denmark (DTU), department of Informatics and Mathematical Modelling (IMM) in partial fulfilment of the requirement for acquiring the PhD degree in engineering.

The topic of the thesis is application of Markov and mixed models to the analysis of sleep EEG data and pharmacokinetic and pharmacodynamic models in general. The thesis consists of a summary report and five research papers written during the PhD study. Four are submitted to or published in international journals and one is published as a research report at DTU/IMM.

I would like to thank my supervisors Henrik Madsen from DTU/IMM and Philip Hougaard from H. Lundbeck A/S for their excellent support and ideas during the project. Also, I wish to thank my colleagues at both DTU/IMM and H. Lundbeck for valuable discussions and for making it a enjoyable time during the project and also in particular my fellow PhD students Anna Helga Jónsdóttír, Søren Klim and Rune H. B. Christensen for their collaboration in different projects.

Lyngby, September 2009

Stig Bousgaard Mortensen

Abstract

This thesis deals with mathematical and statistical models with focus on applications in pharmacokinetic and pharmacodynamic (PK/PD) modelling. These models are today an important aspect of the drug development in the pharmaceutical industry and continued research in statistical methodology within these areas are thus important.

PK models are concerned with describing the concentration profile of a drug in both humans and animals after drug intake whereas PD models are used to describe the effect of a drug in relation to the drug concentration. PK models for an individual are usually described as a deterministic mean value using ordinary differential equations to which a random error is added. This thesis explores methods based on stochastic differential equations (SDEs) to extend the models to more adequately describe both true random biological variations and also variations due to unknown or uncontrollable factors in an individual. Modelling using SDEs also provides new tools for estimation of unknown inputs to a system and is illustrated with an application to estimation of insulin secretion rates in diabetic patients.

Models for the effect of a drug is a broader area since drugs may affect the individual in almost any thinkable way. This project focuses on measuring the effects on sleep in both humans and animals. The sleep process is usually analyzed by categorizing small time segments into a number of sleep states and this can be modelled using a Markov process. For this purpose new methods for non-parametric estimation of Markov processes are proposed to give a detailed description of the sleep process during the night.

Statistically the Markov models considered for sleep states are closely related to the PK models based on SDEs as both models share the Markov property. When the models are applied to clinical data there will often be a large variation between individuals and this can be included in both types of models using the mixed modelling approach. Estimation in these models is discussed with emphasis on data with a more complex grouping structure.

Resumé

(Abstract in Danish.)

Denne afhandling beskæftiger sig med matematiske og statistiske modeller med fokus på applikationer indenfor farmakokinetisk og farmakodynamisk (PK/PD) modellering. Disse modeller udgør i dag et vigtigt aspekt af udvikling af lægemidler i den farmaceutiske industri og fortsat forskning i statistiske metoder inden for disse områder er derfor vigtig.

PK modeller beskriver koncentrationsprofilen af medicin i både mennesker og dyr efter indtagelse af et lægemiddel hvorimod PD modeller bruges til at beskrive virkningen af et lægemiddel i relation til koncentrationsprofilen. Oftest bliver PK modeller for et individ beskrevet deterministisk ved brug af ordinære differentialligninger. Denne afhandling undersøger metoder baseret på stokastiske differentialligninger (SDEer) der gør det muligt udvide disse modeller til at beskrive variation fra både tilfældige biologiske effekter og også variationer fra ukendte eller ukontrollerbare faktorer i et individ. Modellering ved hjælp af SDEer giver også nye metoder til estimering af ukendt input til et system, og dette er illustreret her med en anvendelse til estimering af raten for insulinproduktion i diabetespatienter.

Modeller for virkningen af lægemidler er et bredere område da de kan påvirke det enkelte individ på et næsten ubegrænset antal måder. Dette projekt fokuserer på at måle effekten på søvn i både mennesker og dyr. Søvnstrukturen bliver normalt analyseret ved at kategorisere små tidssegmenter i et antal søvnstadier, og dette kan modelleres ved brug af en Markov proces. Til dette formål er der udviklet nye ikke-parametriske metoder til estimering af Markov processer for at kunne give en detaljeret beskrivelse af søvnstrukturen i løbet af natten.

Statistisk er Markov modeller for søvnstadier tæt relateret til PK modeller baseret på SDEer da de begge deler Markov egenskaben. Når disse modeller anvendes til kliniske data, vil der ofte være en stor variation mellem enkeltpersoner og denne kan beskrives i begge typer af modeller ved at anvende såkaldte mixede modeller. Estimation i disse modeller bliver behandlet med vægt på data med en mere kompleks grupperingsstruktur.

List of publications

The thesis is based on the following five scientific research papers,

A Mortensen SB, Klim S, Dammann B, Kristensen NR, Madsen H, Overgaard RV. A Matlab Framework for Estimation of NLME Models using Stochastic Differential Equations: Applications for estimation of insulin secretion rates. *Journal of Pharmacokinetics and Pharmacodynamics* 34, pp. 623-42 (2007).

B Mortensen SB, Jónsdóttír AH, Klim S, and Madsen H. Introduction to PK/PD modelling with focus on PK and stochastic differential equations. *IMM-Technical Report-2008-16* (2008).

C Klim S, Mortensen SB, Kristensen NR, Overgaard RV and Madsen H. Population stochastic modelling (PSM) - An R package for mixed-effects models based on stochastic differential equations. *Computer Methods and Programs in Biomedicine* 94, pp. 279-289 (2009).

D Mortensen SB, Madsen H, and Hougaard P. Local Estimation of a Discretely Observed Continuous Time Inhomogeneous Markov Jump Process. *Submitted to Statistical Modelling* (2009).

E Mortensen SB and Christensen RHB. Flexible Estimation of nonlinear mixed models via Laplace's approximation. *Submitted to Journal of the American Statistical Association* (2009).

Below is a list of other publications also prepared during the PhD project. The scientific content is covered in papers A, B and C and they will thus not be adressed in this thesis.

- Klim S, Mortensen SB, Madsen H, Overgaard RV and Kristensen NR. "Stochastic PK/PD modelling". Poster for *Séminaire Européen de Statistique*, Cartagena, Spain (2007).
- Mortensen SB og Klim S. "Population Stochastic Modelling (PSM): Model definition, description and examples". Package vignette for the R package PSM. Url: http://www.imm.dtu.dk/psm/doc/PSM.pdf (2008).
- Mortensen SB og Klim S. "Package: PSM". User manual for the R package PSM. Url: http://www.imm.dtu.dk/psm/PSM-help.pdf (2008).
- Klim S, Mortensen SB and Madsen H. Linear Mixed Effects models based on Stochastic Differential Equations in R. Poster for *Annual Meeting of the Population Approach Group in Europe*, Marseille, France (2008).
- Jónsdóttír AH, Klim S, Mortensen SB, and Madsen H. Chapter "Matematik i medicinudvikling" (eng.: Mathematics in drug development) for the book "Matematiske horisonter" (eng.: Mathematical horizons), ISBN 978-87-643-0453-4 (2009).
- Nielsen HB and Mortensen SB. "Package: ucminf" User manual for for the R package ucminf for quasi-Newton optimization. Url: http://cran.r-project.org/web/packages/ucminf/

In collaboration with other researchers the following papers were also prepared during the PhD project. They are based on earlier work and hence will not be adressed here.

- Yoon CH, Bödvarsson B, Klim S, Mørkebjerg M, Mortensen SB, Chen J, Maclaren JR, Luther PK, Squire JM, Bones PJ, Millane RP. Determination of Myosin Filament Orientations in Electron Micrographs of Muscle Cross Sections. *IEEE Trans. Image Process.*, 18(4), 831-839 (2009).
- Bödvarsson B, Klim S, Mørkebjerg M, Mortensen SB, Yoon CH, Chen J, Maclaren JR, Luther PK, Squire JM, Bones PJ and Millane RP. A morphological image processing method for locating myosin filaments in muscle electron micrographs. *Image and Vision Computing* 26, 1073-1080 (2008).

Contents

Preface **i**

Abstract **iii**

List of publications **v**

1 Introduction **1**

2 Mixed models **5**
2.1 Parameter estimation 6
2.2 Evaluation of the marginal likelihood 6
2.3 Multivariate Laplace approximation 9
2.4 The nonlinear mixed model 11

3 Inhomogeneous Markov processes **15**
3.1 Sleep EEG 16
3.2 Model definition 23
3.3 Non-parametric estimation 26
3.4 Parametric estimation 33
3.5 Discussion 36

4 Stochastic differential equations **39**
4.1 Brownian motion 40
4.2 Itô integrals 40

4.3 Filtering problem 42
4.4 Likelihood estimation 45
4.5 Mixed models with SDEs 46
4.6 Applications of SDE based models 47

5 Conclusion **51**

A Paper A **59**

B Paper B **81**

C Paper C **121**

D Paper D **133**

E Paper E **149**

Chapter 1

Introduction

The drug development process in the pharmaceutical industry today generates increasing amounts of data and this requires ongoing development and refinement of the methods applied for the analysis. The present project explores methods based on Markov and nonlinear mixed effects models with applications mainly within pharmacokinetics and pharmacodynamics (PK/PD).

Nonlinear mixed models (NLMMs) are used as a standard tool today for what is often referred to as population PK/PD modelling. Mixed models are models with both fixed and random effects and can in many cases effectively describe both the common structure of a response and the random variation between e.g. a number of individuals in the data. This thesis extends the class of models that can easily be fitted in this framework. For NLMMs the maximum likelihood estimation involves an integral of dimension equal to the number of random effects in the model. In cases where random effects only occur for one level of grouping in the data (e.g. individuals) or if random effects are nested (e.g. separate groups of individuals are observed at different study centers) the dimensionality of the integration problem can be significantly reduced using standard methods. This type of model structure has almost solely been the focus of statistical software for estimation of NLMMs, which goes back to the first software tool NONMEM (Beal and Sheiner, 2004) which was initially introduced around 1980 (Beal and Sheiner, 1980). Situations with crossed random effects where e.g. some individuals are observed at more than one study center requires an evaluation of the full dimensional integral. In Chapter 2 it will be described how this problem can be handled using the multivariate Laplace approximation.

Pharmacokinetics and pharmacodynamics are in popular terms often described as "what the body does to the drug" and "what the drug does to the body", respectively. Research in these disciplines has a long history and today these disciplines constitute an important part of the drug development process. A general introduction to PK/PD is given in paper B and a more thorough description can be found in e.g. Rowland and Tozer (1997) or (Gabrielsson and Weiner, 2006).

Pharmacodynamics is concerned with the effects of the drug. The effect of a drug can e.g. be lowering of the blood pressure, in which case it is relatively straight forward to measure, or it can be reducing pain, which is somewhat more difficult to quantify. An example of a PD model for a pain reliever is illustrated in paper B. Here the effect is measured on a visual analogue scale and this is linked to a PK model to get a full PK/PD model for the drug.

A part of this project has been concerned with effect measures for sleep and in particular a measure of sleep related to time. This study was initiated by a new drug candidate Gaboxadol which has been under development by H. Lundbeck. Studies have shown that Gaboxadol has sleep promoting effects that differs from currently marketed sleep drugs.

Sleep is mainly studied using the electroencephalogram (EEG) which records electrical signals from the neurons in the brains by placing a number of electrodes on the scalp. The electrical signals are sampled at a high frequency of e.g. 100 Hz which results in a raw EEG recording. This recording is traditionally transformed into sleep stages, where each segment of data of usually either 10 or 30 seconds (called the epoch length) is classified into sleep stages. For humans there are six stages, namely wake, REM sleep and sleep stages 1 to 4. These sleep stages were defined by Rechtschaffen and Kales (1968) and has been used as a gold standard ever since. For rodents similar analysis can be performed, but here it is usually only possible to classify 3 sleep stages, namely wake, delta sleep and paradoxical sleep where the two latter are related to sleep stages 3-4 and REM sleep in humans, respectively.

Traditionally sleep has been evaluated by means of simple summaries like time spent in each sleep stage, time to sleep onset and similar measures. In order to study the dynamics of changes of the sleep process over night in more detail it is chosen to model the sequence of sleep stages as an inhomogeneous Markov process. Estimation of changes in parameters of the inhomogeneous Markov model is discussed in Chapter 3 where a method based on local kernel estimation is proposed for solving the estimation problem. It is further shown how this can be used to estimate the pharmacodynamic effect of Gaboxadol.

Pharmacokinetics is the study of how a drug enters and circulates through the body and in general it describes where the drug is in the body as a function of time. The dynamical description of the body as a system is obtained using a compartmental model structure, where each compartment represents an area

of the body where the drug is contained and can be assumed to be evenly distributed. Examples of potential compartments are the bloodstream, muscles, the stomach or peripheral tissue. The movement of the drug between the compartments is traditionally described using ordinary differential equations and thus implicitly defines the body as a deterministic system without any random biological variation. If such random biological variation is actually present or if some parts of the system is not adequately described this may result in serial correlation in the residuals between observations and model in a statistical analysis.

To appropriately model these phenomena is has been proposed to introduce stochastic differential equations (SDEs) for the modelling. By using SDEs it is possible to include stochastic components in the model of a system to compensate for true biological variation or structural misspecification of the model. A part of this project has been concerned with the development of a software tool for estimation of the embedded parameters of a general nonlinear mixed model based on SDEs. This will be presented and discussed in more detail in Chapter 4 and will also include applications of the modelling approach. Like for the mixed model this work allows for using models that were previously too complex to be used.

Chapter 2

Mixed models

Mixed models make up a general class of models for the analysis of grouped data. A mixed model handles dependence between observations within groups by assuming the existence of one or more unobserved latent variables for each group of data. The unobserved latent variables are assumed to be random and hence referred to as random effects. A mixed model thereby consists of both fixed model parameters $\boldsymbol{\theta}$ and random effects $\boldsymbol{b}$ for modelling both the common and group dependent structures in the data. Modelling using both fixed and random effects has coined the term mixed-effects models or in short just mixed models.

Random effects naturally enter the modelling when data is observed based on a number of experimental units that are taken from a larger population. In such cases it is often found that e.g. individuals from a population are very similar but exhibit a certain amount of variation that is most appropriately modelled as random. The distribution of the random effects thus represents the random variation in the populations that cannot otherwise be reasonably explained by fixed effects and covariates. In this way inference based on a mixed model where the random effect distribution has been estimated along with the model parameters will apply to the population as a whole and not only the individuals selected for the study. From an inferential point of view this is the main benefit achieved using a mixed modelling approach along with the possibility of judging individual covariates such as sex, weight and height.

There are no constraints on the assumed distribution for the random effects, but for a very wide range of applications they are assumed to have a Gaussian

distribution with mean zero such that

$$\boldsymbol{b} \sim N(\mathbf{0}, \boldsymbol{\Psi}).$$

The random effect distribution is thus completely described by its covariance matrix $\boldsymbol{\Psi}$ and this class of Gaussian mixed models will be the focus here.

2.1 Parameter estimation

Estimation in mixed models is based on maximum likelihood. The statistical description of a mixed model is defined by a joint likelihood of the model parameters and the unobserved random effects based on the joint density of $(\boldsymbol{y}, \boldsymbol{b})$. The joint likelihood is given as

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) &= p(\boldsymbol{y}, \boldsymbol{b}|\boldsymbol{\theta}, \boldsymbol{\Psi}) && (2.1)\\ &= p(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{\Psi})p(\boldsymbol{b}|\boldsymbol{\theta}, \boldsymbol{\Psi}) && (2.2)\\ &= p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta})p_2(\boldsymbol{b}|\boldsymbol{\Psi}) && (2.3) \end{aligned}$$

where (2.2) follows using Bayes' theorem and (2.3) since $p_1(\boldsymbol{y}|\boldsymbol{b})$ does not involve $\boldsymbol{\Psi}$ and $p_2(\boldsymbol{b}|\boldsymbol{\Psi})$ likewise does not involve $\boldsymbol{\theta}$. The part of the model defining $p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta})$ is sometimes referred to as the first stage model (with likelihood function $L_1(\boldsymbol{b}, \boldsymbol{\theta}|\boldsymbol{y})$) and $p_2(\boldsymbol{b}|\boldsymbol{\Psi})$ as the second stage model. To obtain the likelihood for the model parameters $(\boldsymbol{\theta}, \boldsymbol{\Psi})$ the unobserved random effects are integrated out based on the assumed (here Gaussian) distribution. The likelihood function for estimating $(\boldsymbol{\theta}, \boldsymbol{\Psi})$ is thus the marginal likelihood

$$L_M(\boldsymbol{\theta}, \boldsymbol{\Psi}|\boldsymbol{y}) = \int_{\mathbb{R}^q} L(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) d\boldsymbol{b} \tag{2.4}$$

where q is the number of random effects and $\boldsymbol{\theta}$ and $\boldsymbol{\Psi}$ are the parameters to be estimated. The likelihood function in (2.4) gives a very broad definition of mixed models: the only requirement for using mixed modeling is to define a joint likelihood function for the model of interest. In this way mixed modelling can be applied to any likelihood based statistical modelling. Examples of applications are linear mixed models (LMM) and nonlinear mixed models (NLMM) (regression type models, see e.g. Bates and Watts (1988)), generalized linear mixed models (McCulloch and Searle, 2001) but also models based on Markov chains and SDEs as will be the focus of Chapter 3 and 4.

2.2 Evaluation of the marginal likelihood

For many classes of statistical models the integral in (2.4) is intractable with no closed form solution available. An important exception to this is the widely

used Gaussian linear mixed model where $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b} + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$, for which the marginal distribution of the observation $\boldsymbol{y}$ is given explicitly as a multivariate Gaussian distribution with mean $\boldsymbol{X}\boldsymbol{\beta}$ and variance $\boldsymbol{Z}\boldsymbol{\Psi}\boldsymbol{Z}^T + \boldsymbol{\Sigma}$. For Gaussian nonlinear mixed models

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{\beta}, \boldsymbol{b}) + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{\Sigma}), \tag{2.5}$$

where the model function $\boldsymbol{f}$ is nonlinear, it is no longer generally possible to derive an explicit marginal distribution. An exception to this is nonlinear mixed models which are nonlinear only in $\boldsymbol{\beta}$ but not $\boldsymbol{b}$. Such a model can be re-written with a first-order Taylor expansion around $\boldsymbol{b} = \boldsymbol{0}$ as $\boldsymbol{f}(\boldsymbol{\beta}, \boldsymbol{b}) = \boldsymbol{f}(\boldsymbol{\beta}, \boldsymbol{0}) + \boldsymbol{f}'_{\boldsymbol{b}}(\boldsymbol{\beta})\boldsymbol{b}$ which is equivalent to a linear mixed model with $\boldsymbol{X}\boldsymbol{\beta} = \boldsymbol{f}(\boldsymbol{\beta}, \boldsymbol{0})$ and $\boldsymbol{Z}\boldsymbol{b} = \boldsymbol{f}'_{\boldsymbol{b}}(\boldsymbol{\beta})\boldsymbol{b}$. For a given set of parameters $\boldsymbol{\beta}$ the likelihood function can thus be evaluated based on the marginal multivariate Gaussian distribution as in the linear mixed model. A clear distinction between nonlinear mixed models that are either linear or nonlinear in the random effects is not always made, but is important from a computational viewpoint.

2.2.1 Likelihood approximations

For mixed models where there is no closed form solution to (2.4) it is necessary to invoke some form of numerical approximation to be able to estimate the model parameters. The complexity of this problem is mainly dependent the dimensionality of the integration problem which in turn is dependent on the grouping structure in the data for the random effects. These structures include a single grouping, nested grouping, partially crossed and crossed random effects.

For problems with only one level of grouping the marginal likelihood can be simplified as

$$L(\boldsymbol{\beta}, \boldsymbol{\Psi}|\boldsymbol{y}) = \prod_{i=1}^{M} \int_{\mathbb{R}^{q_i}} p_1(\boldsymbol{y}|\boldsymbol{b}_i, \boldsymbol{\theta}) p_2(\boldsymbol{b}_i|\boldsymbol{\Psi}) d\boldsymbol{b}_i \tag{2.6}$$

where q_i is the number of random effects for each group and M is the number of groups. Instead of having to solve an integral of dimension q it is only necessary to solve M smaller integrals of dimension q_i. In typical applications there is often just one or only a few random effects for each group, and this thus greatly reduces the complexity of the integration problem. If the data has a nested grouping structure a reduction of the dimensionality of the integral similar to that shown in (2.6) can be performed. An example of a nested grouping structure is data collected from a number of schools, a number of classes within each school and a number of students from each class. However, if some students changes school during the study, the random effects structure is suddenly partially crossed and the simplification relating to (2.6) no longer applies.

Estimation in models with a single level of grouping has received the main focus of research within nonlinear mixed models and a number of approxima-

tions have been proposed in the literature for the marginal likelihood function in (2.6). Pinheiro and Bates (1995) compare the five most common methods, namely the Lindstrom and Bates alternating method, a modified Laplacian approximation, importance sampling, Gaussian quadrature and adaptive Gaussian quadrature. They conclude that the Lindstrom and Bates alternating method performs well and is most computationally efficient. This method is however only applicable for data with a single or nested grouping structure and further requires individual parameters to be modelled as a linear function of the random effects $\boldsymbol{\phi}_{ij} = \boldsymbol{A}_{ij}\boldsymbol{\beta} + \boldsymbol{B}_{ij}\boldsymbol{b}_i$ where $\boldsymbol{A}_{ij}$ and $\boldsymbol{B}_{ij}$ are design matrices for the individual parameters and this constrains the individual parameters $\boldsymbol{\phi}_{ij}$ to be normally distributed. This limits the alternating method from applications such as pharmacokinetics, where log-normally distributed parameters are often encountered.

Pinheiro and Bates further conclude that the Laplacian or adaptive Gaussian approximations should be used when greater accuracy is required. The Laplacian approximation is equivalent to adaptive Gaussian with one quadrature point, and further points can thus be used in adaptive Gaussian to improve the Laplacian approximation. However, increasing the number of points gave only marginal improvement and did not seem necessary. Importance sampling can be used to achieve similar accuracy, but was found to be less computationally efficient and also the stochastic element can give problems for gradient based estimation procedures of the model parameters. Gaussian quadrature were found to be too inefficient due to poor sampling of the integrand.

Both importance sampling, the Laplacian and adaptive Gaussian approximations are exact when the random effects are linear in the random effects. An extension of this is a model where some random effects are linear and some are nonlinear which is discussed in du Toit and Cudeck (2009). If there is only one level of grouping they show that it is possible to separate these and only use a numerical integration method for the integration over the nonlinear random effects.

If the nonlinear mixed model is extended to include any structure of random effects such as crossed or partially crossed random effects it is required to evaluate the full multiple integral in (2.4) as mentioned earlier. This may significantly increase computational demands due to the product rule. This states that if an integral is sampled in m points per dimension to evaluate it, the total number of samples needed is m^q which rapidly becomes infeasible even for a limited number of random effects. For this reason estimation in models with crossed random effects is not supported by any of the standard software packages for fitting NLMMs such as nlme (Pinheiro et al., 2008), SAS NLMIXED (SAS Institute Inc., 2004) and NONMEM (Beal and Sheiner, 2004). However, estimation in these model can efficiently be handled using the multivariate Laplace approximation, which only samples the integrand in one point common to all dimensions. Estimation based on the multivariate Laplace approximation is the

main focus for paper E where it is shown how it can be implemented on a case by case basis in R (R Development Core Team, 2008) with a limited amount of coding required to make it possible to estimate NLMMs with any structure of random effects.

2.3 Multivariate Laplace approximation

The Laplace approximation will be outlined in the following with application to other than standard Gaussian nonlinear mixed models in mind. A thorough description of the Laplace approximation in nonlinear mixed models is found in Wolfinger and Lin (1997) and it is also presented in paper E. In short, the Laplace approximation approximates the joint likelihood in (2.3) with a Gaussian distribution centered at the modes of the random effects. This corresponds to a 2nd order Taylor expansion (i.e. a quadratic approximation) of the joint log-likelihood h given as

$$h(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) = \log L(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) \tag{2.7}$$
$$= \log p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta}) + \log p_2(\boldsymbol{b}|\boldsymbol{\Psi}) \tag{2.8}$$

where the expansion of h is done around the mode of the random effects $\tilde{\boldsymbol{b}} = \arg\max_{\boldsymbol{b}} L(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y})$ for fixed $\boldsymbol{\theta}$ and $\boldsymbol{\Psi}$. Note that in paper E the parameter $\boldsymbol{\theta}$ includes $\boldsymbol{\Psi}$, but here it is noted separately to distinguish the parameters used only in the first stage model. The second-order Taylor expansion of the joint log-likelihood h is given as

$$\log L(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) = h(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b}|\boldsymbol{y}) \tag{2.9}$$
$$\approx h(\boldsymbol{\theta}, \boldsymbol{\Psi}, \tilde{\boldsymbol{b}}|\boldsymbol{y}) - \frac{1}{2}(\boldsymbol{b} - \tilde{\boldsymbol{b}})^T \boldsymbol{H}(\tilde{\boldsymbol{b}})(\boldsymbol{b} - \tilde{\boldsymbol{b}}) \tag{2.10}$$

where the first-order term in the Taylor expansion disappears since the expansion is done around the mode $\tilde{\boldsymbol{b}}$ and $\boldsymbol{H}(\tilde{\boldsymbol{b}}) = -h''_{\boldsymbol{bb}}(\boldsymbol{\theta}, \boldsymbol{\Psi}, \boldsymbol{b})|_{\boldsymbol{b}=\tilde{\boldsymbol{b}}}$ is the *negative* Hessian of the joint log-likelihood evaluated at $\tilde{\boldsymbol{b}}$ which will simply be referred to as 'the Hessian'. Using the approximation in (2.10) in (2.4) the Laplace

log-likelihood can be derived as

$$\begin{aligned}
\log L_{LA}(\boldsymbol{\theta}, \boldsymbol{\Psi}|\boldsymbol{y}) &= \log \int_{\mathbb{R}^q} \exp\left(h(\boldsymbol{\theta}, \boldsymbol{\Psi}, \tilde{\boldsymbol{b}}|\boldsymbol{y}) - \frac{1}{2}(\boldsymbol{b}-\tilde{\boldsymbol{b}})^T \boldsymbol{H}(\tilde{\boldsymbol{b}})(\boldsymbol{b}-\tilde{\boldsymbol{b}}) \right) d\boldsymbol{b} \\
&= h(\cdot) + \log \int_{\mathbb{R}^q} \exp\left(-\frac{1}{2}(\boldsymbol{b}-\tilde{\boldsymbol{b}})^T \boldsymbol{H}(\tilde{\boldsymbol{b}})(\boldsymbol{b}-\tilde{\boldsymbol{b}}) \right) d\boldsymbol{b} \\
&= h(\cdot) + \log \left| \frac{2\pi}{\boldsymbol{H}(\tilde{\boldsymbol{b}})} \right|^{\frac{1}{2}} \int_{\mathbb{R}^q} \frac{1}{(2\pi)^{\frac{q}{2}} |\boldsymbol{H}^{-1}(\tilde{\boldsymbol{b}})|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{b}-\tilde{\boldsymbol{b}})^T \boldsymbol{H}(\tilde{\boldsymbol{b}})(\boldsymbol{b}-\tilde{\boldsymbol{b}}) \right) d\boldsymbol{b} \\
&= h(\cdot) + \log \left| \frac{2\pi}{\boldsymbol{H}(\tilde{\boldsymbol{b}})} \right|^{\frac{1}{2}} \\
&= h(\boldsymbol{\theta}, \boldsymbol{\Psi}, \tilde{\boldsymbol{b}}|\boldsymbol{y}) - \frac{1}{2} \log \left| \frac{\boldsymbol{H}(\tilde{\boldsymbol{b}})}{2\pi} \right| \qquad (2.11) \\
&= \log p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta}) - \frac{1}{2} \log |2\pi\boldsymbol{\Psi}| - \frac{1}{2} \tilde{\boldsymbol{b}}^T \boldsymbol{\Psi}^{-1} \tilde{\boldsymbol{b}} - \frac{1}{2} \log \left| \frac{\boldsymbol{H}(\tilde{\boldsymbol{b}})}{2\pi} \right| \\
&= \log p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta}) - \frac{1}{2} \log |\boldsymbol{\Psi}| - \frac{1}{2} \tilde{\boldsymbol{b}}^T \boldsymbol{\Psi}^{-1} \tilde{\boldsymbol{b}} - \frac{1}{2} \log |\boldsymbol{H}(\tilde{\boldsymbol{b}})| \qquad (2.12)
\end{aligned}$$

where the integral is eliminated by transforming it to an integration of a multivariate Gaussian density with mean $\tilde{\boldsymbol{b}}$ and variance $\boldsymbol{H}^{-1}(\tilde{\boldsymbol{b}})$. In the step in (2.11) the fraction in the determinant is inverted to avoid a matrix inversion of the Hessian. In (2.12) the log-likelihood function is written out further to show the different components and it is slightly simplified by using that the constant $-q \log 2\pi$ from the determinant of the Hessian cancels out with $q \log 2\pi$ from the density function for the random effects.

The evaluation of the Laplace likelihood in (2.12) makes no assumptions on the first stage model $L_1(\boldsymbol{b}, \boldsymbol{\theta}|\boldsymbol{y}) = p_1(\boldsymbol{y}|\boldsymbol{b}, \boldsymbol{\theta})$. As long as a likelihood function of the random effects and model parameters can be defined it is possible to use the Laplace likelihood for estimation in a mixed model framework. This will be discussed further with the models used in Chapter 3 and 4.

The Laplace likelihood only approximates the marginal likelihood for mixed models with nonlinear random effects and thus maximizing the Laplace likelihood will result in some amount of error in the resulting estimates. However, in Vonesh (1996) it is shown that the joint log-likelihood converges to a quadratic function of the random effect for increasing number of observations per random effect and thus that the Laplace approximation is asymptotically exact. When choosing to use a Gaussian distribution for the random effects this can be expected to give a faster rate of convergence compared to other distributions, since the logarithm of the Gaussian distribution is in itself quadratic in the random effects as seen in (2.12). In situations where individual random effects are relatively well defined the Laplace approximation can thus be expected to per-

form well and this holds independently of the first stage model. This makes the Laplace approximation an attractive approach for many applications. However, in practical applications the accuracy of the Laplace approximation may still be of concern, but improved numerical approximation of the marginal likelihood (such as Gaussian quadrature) may often easily be computationally infeasible to perform. This concern is addressed in paper E where it is suggested to use graphical methods to asses the accuracy of the approximation.

2.4 The nonlinear mixed model

As mentioned previously, the estimation scheme presented in paper E is based on the multivariate Laplace approximation in (2.12) to give full flexibility in random effects structure not otherwise found in standard software. The paper focuses on the standard nonlinear mixed model with first stage model given by (2.5) and the main point is to illustrate how an NLMM can be defined and estimated in R with relative ease.

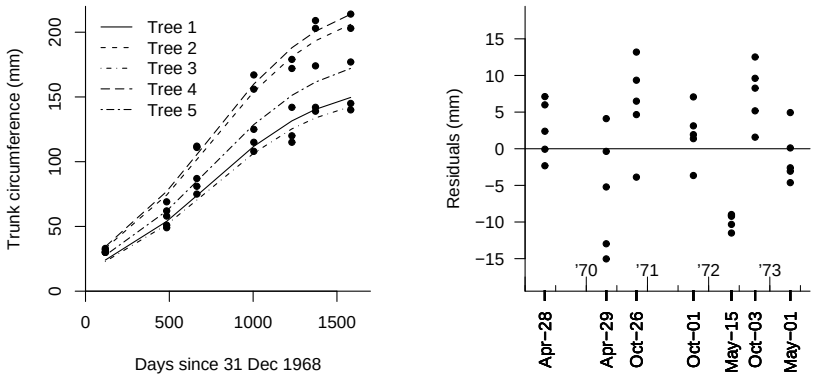
In the NLMM it is possible to apply a first order approximation of the Hessian in (2.10) when using the Laplace approximation. This approximation is equal to the expected Hessian (in both linear and nonlinear models) and is referred to as the *modified* Laplace approximation by Pinheiro and Bates (1995) and as the FOCE approximation in NONMEM (Wang, 2007). Further details are found in paper E.

To illustrate the estimation scheme presented in paper E a data set with the growth of 5 orange trees is used (Draper and Smith, 1981). The circumferences of the trees are measured 7 times approximately every half year over a 4 year period. The data has been used previously in the literature by Lindstrom and Bates (1990) (see Figure 2.1) and Millar (2004) for illustration of NLMMs. The former uses a single random effect for difference in asymptotic circumference and the latter introduces an additional crossed random effect to handle difference between sampling occasions.

In paper E it is shown that the apparently random effect of the sampling occasion can in fact be explained by a deterministic effect of season and the fact that some half year intervals are missed making the effect look 'random'. It is also found that residuals within trees are strongly correlated in time which can be modelled using a continuous auto-regressive (CAR) model given as

$$\mathrm{cov}(\epsilon_{ij}, \epsilon_{ik}) = \sigma^2 \exp(-\phi|t_{j'} - t_j|) \tag{2.13}$$

where t_j is time in days for the jth observations of the ith tree. The resulting model is similar to a model proposed for repeated measurements of growth of rats suggested by Diggle (1988) which includes a random effect for variation between rats and serial correlation. Although a continuous AR model is chosen here, a similar model could be achieved using a discrete AR model where the correlation



(a) Orange tree data and model fit.

(b) Residuals versus time.

Figure 2.1: Plots of orange tree data together with the fit of a model by Lindstrom and Bates (1990).

between measurements within a tree is ϕ^k, where k denotes the number of half year periods between two observations and thereby correctly handling the missed sampling occasions (Madsen, 2008). A more detailed interpretation of the correlation structure and its relation with stochastic differential equations will be discussed further in Chapter 4.

From an estimation perspective the model with crossed random effects (possibly in combination with the CAR model) is of particular interest, as these cannot be estimated using standard software. In paper E it is shown how all the models discussed are efficiently estimated using the suggested framework for estimation in R.

2.4.1 Likelihood based inference

The estimation scheme for NLMMs in R gives the extra advantage of having the marginal log-likelihood available in R. This can be used to directly create profile likelihood for parameters in the model, which is not easily possible in other standard software packages.

Profile likelihood plots are a key element in likelihood based inference since it contains all information about to what extent different values of a parameter are supported by the data. The profile likelihood can be used to make likelihood based confidence intervals (CIs) instead of having to rely on the standard Wald CI which is based on a quadratic approximation of the log-likelihood function. The Wald CI is meaningful only if $\log L(\theta)$ is at least approximately quadratic and if this is not the case it is necessary to find a normalizing transform. The

likelihood confidence interval is superior to the Wald approximation in the sense that it automatically employs the best possible normalizing transform without needing to know it. The likelihood CI is thus always as good as or better than the Wald CI and will thus also better approximate the advertised coverage probability (Pawitan, 2001).

These well established aspects of likelihood based inference touch upon an important issue of how to make inference based on collected data in general. This has been the focus of much debate and controversy through out the history of statistics. It is still a highly relevant research area and thus deserves some discussion.

Traditionally statistical results have been reported using either Fisher's p-value for measuring evidence against a null-hypothesis or using the Neyman-Pearson hypothesis testing for choosing between a null and alternative hypothesis using a decision rule that controls the long term error rates (Blume and Peipert, 2003). Although the two approaches are fundamentally different in objective they are numerically closely related and this has led to some confusion. If a researcher chooses a significance level $\alpha = 0.05$ and finds $p = 0.0003$ after conducting the study he can with confidence act as if the alternative hypothesis is true with assurance given by the long term error rates. However, as a researcher he might also *at the same time* argue that the small p-value provides evidence against the null hypothesis in the particular study at hand (as argued by Fisher). This is wrong however: a single number cannot both be seen from a short and long run perspective. More detailed arguments for this can be found in Goodman (1999).

A part from the confusion caused by the mix of the two methods, none of them serve typical research purposes well. General research is not a matter of making decisions and also the p-value is easily misleading since it does not reveal any information about the range of effect sizes supported by data. This has led to a greater focus on reporting results using confidence intervals, profile likelihood and other likelihood based methods (Royall, 1997) as these methods more adequately convey the statistical evidence available in the data. The estimation framework presented in paper E supports these ideas by making further analysis of the likelihood function in mixed models directly available.

2.4.2 Computational aspects

The computational complexity of using the mixed modelling framework will always be somewhat greater than using the first stage model alone or by e.g. estimating a set of common parameters across individuals simply using a pooled likelihood. By pooled likelihood for data from M individuals $\boldsymbol{y}_1, ..., \boldsymbol{y}_M$ is meant the likelihood function $L(\boldsymbol{\theta}) = \prod_{i=1}^{M} L(\boldsymbol{\theta}|\boldsymbol{y}_i)$. With estimation in a mixed model using the Laplace approximation it is for every evaluation of the marginal like-

lihood required to find the mode of the random effects $\tilde{\boldsymbol{b}}$ and the Hessian for a set of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\Psi}$. This gives rise to a nested optimization structure where the outer optimization of $\boldsymbol{\theta}$ and $\boldsymbol{\Psi}$ involves repeated optimizations of the random effects.

There is usually a number of finite difference approximations involved in the Laplace likelihood, namely for the Hessian and the gradient of the joint and marginal likelihoods used in the optimization of random effects and model parameters. These finite difference may be both time consuming and hinder fast convergence if they are not sufficiently accurate. Recent work by Skaug and Fournier (2006) points to a possible solution to this problem. They present a software package in the C++ program language called ADMB-RE for mixed models based on the Laplace approximation in (2.12). The package supports estimation of mixed modelling for any programmable first stage model. Estimation in the package uses a technique from computer science known as "automatic differentiation" (AD) (Griewank, 2000). This is a technique that exploits the chain rule of calculus to evaluate derivatives of functions defined in computer programs to machine precision. This avoids the inaccuracies using finite difference approximations and also the limitations using symbolic differentiation since any programmable function can be used as input. It can be shown that the gradient of any function can be evaluated with so-called *reverse mode* automatic differentiation in less than four times the cost of evaluating the function itself independently of the number of model parameters. This is a strong result and should be seen in contrast to finite difference approximations, where the cost is proportional to the number of parameters.

There are some limitations of this approach however; model parameters are not allowed to enter in if-statements in the model and the whole stack of operations needed to evaluate the function must be kept in memory. In particular, the latter restricts models from using large systems of differential equations, as the solution may easily involve too many computations to be kept in memory. With these limitations in mind, the estimation in mixed models using AD offers a number of benefits as both gradients of model parameters and random effects and the Hessian can be evaluated to machine precision. Also recently the software has been made freely available as open souce (ADMB Project, 2009). The downside is that the use of the ADME-RE requires the first stage model to be coded in C++ and in general requires a considerable coding experience to work with. For these reasons it has still not gained widespread usage, but it may be worth considering for complex estimation problems.

Chapter 3

Inhomogeneous Markov processes

A Markov process is a stochastic process where all information on the past relevant for predicting future states is given by the current state alone. The process is thus independent of how the process arrived at the current state and how long it has remained there. This property is called the Markov property and is named after the Russian mathematician Andrey Markov (1856-1922) who was the first to study such processes.

For a stochastic process it holds that information about the future state of the process is described in probabilistic terms as conditional probabilities. The state space for the stochastic process may be either continuous or discrete and the process may evolve in either continuous or discrete time. Markov processes in continuous time with continuous state space can for a certain class of these models be described using stochastic differential equations and this will be the topic of Chapter 4. This chapter will focus on Markov processes with a discrete state space and both discrete and continuous time versions will be discussed. When it is necessary to differentiate, the discrete state process in continuous time is referred to as a Markov jump process and in discrete time simply as a Markov chain.

Markov processes are divided into two further broad categories; it can be either time homogeneous or time inhomogeneous. For a homogeneous Markov chain the transition probabilities over a fixed time interval is independent of time which is not required for an inhomogeneous Markov chain. In this chapter

it will be shown how an inhomogeneous Markov jump process can be used as a model for sleep stages and how time changes of parameters defining the inhomogeneous model can be estimated efficiently using a method based on local kernel estimation.

3.1 Sleep EEG

The motivational application for the project is the study of sleep EEG. An electroencephalogram (EEG) records electrical activity of the brain's surface through electrodes placed on designated sites on the scalp. It can be used on both humans and animals to study the activity of the brain and it can be used both for wake and sleeping subjects, but the application here is focused only on the sleep EEG. The frequency content of a recorded sleep EEG signal usually varies from 1 to 30 Hz and the amplitude of the signal ranges from 20μV to 100μV (Forehand, 2003). The frequency and amplitude varies during the night and based on the frequency range and amplitude of the wave they are denoted as delta (0.5-4Hz), theta (4-7Hz), alpha (8-13Hz), and beta (13-30Hz) waves with the highest amplitude for theta waves and the lowest for beta waves. When the brain is active there is mainly high frequency content, whereas inactivity results in a synchronized pattern of low frequency.

Based on the types of waves and other events in the sleep EEG the signal can be classified into stages relating to 'the state of consciousness' with is typically done for epochs of 10 to 30 seconds. There are two general types of sleep states called REM (rapid eye movement) sleep and NREM (Non REM) sleep. For humans NREM sleep is further categorized into four sub-states denoted sleep states I through IV. State I is a transitional state between wake and state II, which is the first true sleep state. States III and IV are the deep sleep stages, often collectively denoted slow wave sleep (SWS) where there is mainly delta activity in the signal. REM sleep is distinctively different from NREM sleep as the EEG signal shows high activity resembling wake. The body is relatively paralyzed during REM sleep with low muscle tone except for the occurrence of rapid eye movements. REM sleep is found in most mammals and is thought to be important for learning and is also the time during which dreams occur (Brodal, 2001). It is sometimes referred to as paradoxical sleep due the seeming contradictions in its characteristics.

Human sleep cycles through the NREM sleep stages and back to REM sleep about every 90 minutes whereas rats can go from wake to sleep and back within minutes. This more fractured sleeping pattern for smaller animals is likely a natural effects of the fact that they need to stay more alert during sleep. In humans the sleep structure changes with age and also show individual differences (Brodal, 2001).

The concept of dividing sleep into a number of states for a certain epoch

length was first developed for humans and was standardized in Rechtschaffen and Kales (1968) manual for scoring sleep. The time series data containing a scored sleep stage for each epoch is called a hypnogram. Traditional analysis of hypnograms to study sleep is done by using a range of standard summary statistics such as total sleep time, latency to persistent sleep, wake after sleep onset, number of awakenings etc. Such summary statistics may be sufficient to show an effect of a drug by improving one or more of these measures, but they do not give a detailed picture of changes in sleep structure during the night. In the following it will be suggested to use new model based methods for describing the time variations of the sleep structure.

3.1.1 Model assumptions and estimation

In order to study the dynamic changes of the sleep process during the night it is chosen to model the sequence of sleep stages as an inhomogeneous Markov process as has also been proposed earlier in the literature (Zung et al., 1965). If isolated periods of time homogeneity is considered, the Markov assumption implies that the time between state transitions (holding times) is exponentially distributed since the probability of leaving the state is constant for every small time step. This has been found to be a reasonable assumption, see e.g. Kemp and Kamphuisen (1986).

Estimation of changes in parameters of the inhomogeneous Markov model has previously been done by binning the data for small time intervals and using standard maximum likelihood estimators for homogeneous Markov processes. This is further developed in this chapter where a method based on local kernel estimation is proposed for the estimation problem. This defines the estimation problem in a well known statistical framework and it will be shown how it can be used to efficiently extract information on the time course of pharmacodynamic effects on sleep.

3.1.2 Data from sleep study on rats

The modelling approach for sleep using Markov processes will be illustrated using a data set from a pre-clinical study on 6 rats weighing 275-300g and housed singly under a 12:12h light:dark cycle with free access to food and water (Anderson et al., 2006).

The study was performed to investigate the sleep effects of Gaboxadol, which is a sleep promoting compound that has been under development by H. Lundbeck (Wafford and Ebert, 2006). The drug has been found to have positive effects on the sleep structure such as increasing the amount of slow wave sleep during the night (Walsh et al., 2007). The clinical development of Gaboxadol as a sleep drug was stopped in 2007 due to discoveries of significant negative side effects.

The 6 rats are each observed for two 23.5 hour periods. Three of the rats rats are treated with placebo in the first period and an oral dose (PO) of 20μg/g Gaboxadol at the beginning of the second period. The other three rats are treated in the reverse order and thus all rats are observed for both a placebo and treatment period. During the first 12 hours the light is kept on, and in the remaining time the light is turned off. Rats are most active in the dark, and the study design thereby resembles a human taking a sleep drug before bed time.

The sleep cycle of the rats is monitored using EEG. The EEG signal is measured with two electrodes implanted in the rat skull and the signal is transmitted using a telemetry device so that the rats can move freely without any wires attached. Based on the EEG three states are classified, namely wake (W), delta sleep (DS) and paradoxical sleep (PS). The DS state corresponds to NREM sleep in humans. These states are determined every 10 seconds giving 8,460 equidistantly spaced observations for each rat. An example of data from one rat during the first 12 hours is shown in Figure 3.1 for both active drug and placebo treatment. This figure corresponds to the hypnogram for the rat and contains the time series of observed sleep states that will be modelled using a Markov process.

The concentration PK profile of Gaboxadol in rats is also sampled in both the brain and blood plasma. Due to the experimental complexity this cannot be done on the same rats that are also used for the sleep EEG measurements and thus only the mean profiles will be used. The mean PK profile data is shown in Table 3.1. The data for concentration in plasma is based on 4 rats and the data for concentration in the brain is based on 5 rats.

Plasma		Brain	
Time	Conc.	Time	Conc.
Hours	μg/ml	Hours	μg/ml
0.0	0.000	0.0	0.000
0.25	3.850	0.5	0.350
0.5	2.826	1.0	0.376
1.0	0.854	1.5	0.398
2.0	0.383	2.0	0.201
4.0	0.179	2.5	0.211
5.0	0.036	3.0	0.176
7.0	0.025	3.5	0.196
		4.0	0.146
		4.5	0.110

Table 3.1: PK data for a 20μg/g oral dose of Gaboxadol in rats.

The PK profiles will be used to compare with the pharmacodynamic effect of Gaboxadol found using the Markov model for the sleep EEG data. In the following a combined PK model for both the plasma and brain concentration profiles is presented. The model is a two compartment model with blood plasma rep-

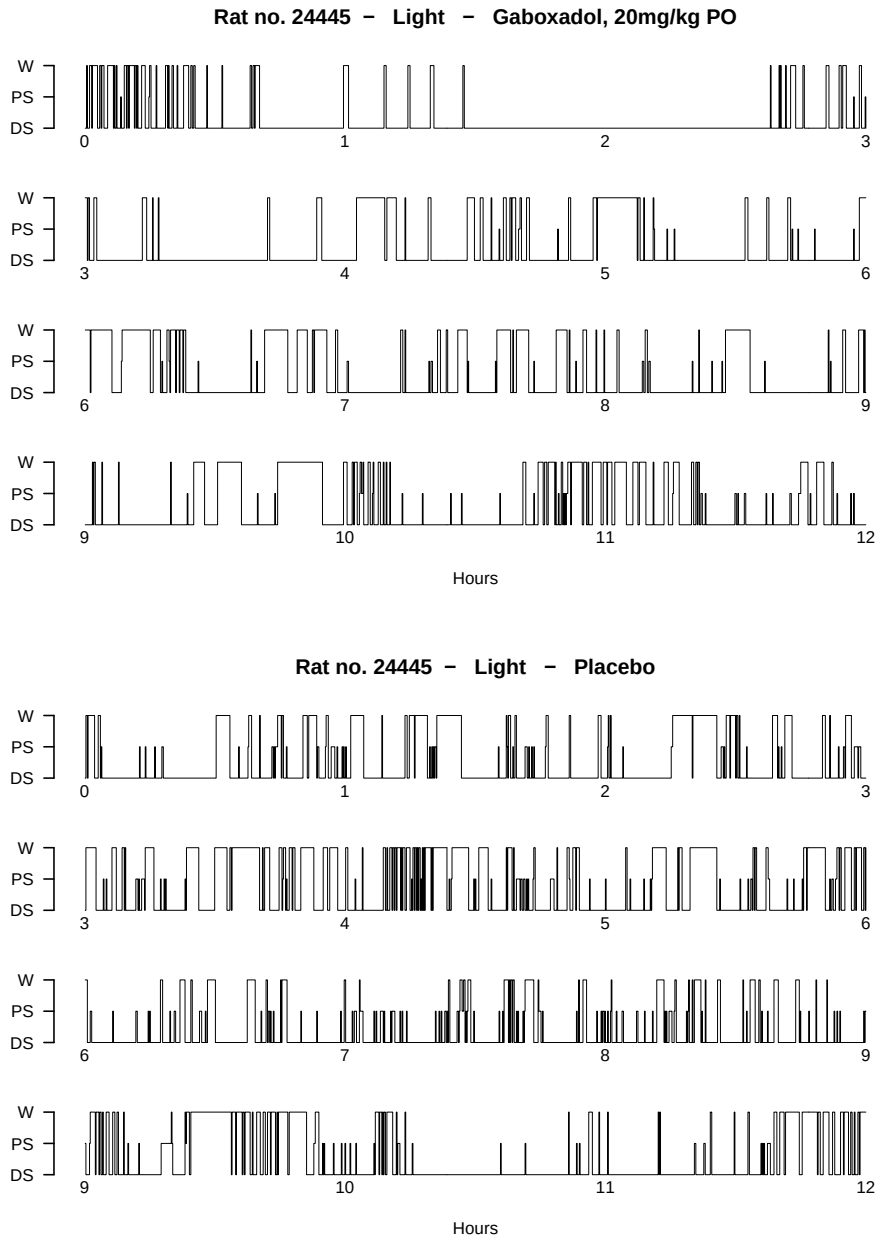


Figure 3.1: Scored sleep states for a rat during the first 12 hours of the study. Active and vehicle drug are shown above and below, respectively.

resented by the central compartment and brain in the peripheral compartment. The orally dosed drug is assumed to be absorbed in the plasma through a first order process. The only route of elimination is through the central compartment which gives the plasma concentration a double exponential decay profile. The model is illustrated in Figure 3.2 and the corresponding model for the mass transfer of Gaboxadol in the system is given as

$$\begin{aligned} dA_s/dt &= -k_a A_s \\ dA_p/dt &= k_a A_s - (k_e + k_{12})A_p + k_{21} A_b \\ dA_b/dt &= k_{12} A_p - k_{21} A_b. \end{aligned} \tag{3.1}$$

The unit for the compartments is $[A] = \mu$g/g which is understood as amount of drug per gram rat since the dose is weight normalized and the units for the rate constants are $[k] = \text{min}^{-1}$.

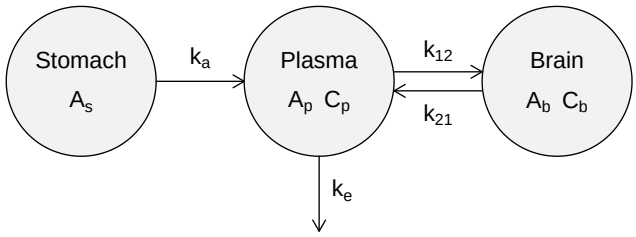


Figure 3.2: PK model for Gaboxadol concentration profiles.

The observations are assumed to be measured with a log-Gaussian distribution around the median response and, that is, the model residuals are additive Gaussian on the log-scale. The number of parameters is limited by assuming equal measurement variance for both plasma and brain concentrations (a residual analysis of the final model fit indicates that this is a reasonable simplification). The volume of distribution for the measurement compartments are denoted V_p and V_b for plasma and brain respectively with a weight normalized unit of $[V] = \text{ml/g}$ due to the weight normalized specification of the dose. This gives the observations the correct unit of $[C] = \mu$g/ml. The measurement equations are given as

$$\begin{aligned} C_p &= A_p/V_p \exp(\epsilon_p), \\ C_b &= A_b/V_b \exp(\epsilon_b). \end{aligned} \tag{3.2}$$

where $[\epsilon_p \ \epsilon_b]^T \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I})$. An additive error model on the original scale for the concentrations was also tried, but since the range of the concentration values is relatively large (3.85 to 0.03 μg/ml) this was found to give too large standardized residuals for the observations in the top of the range and the log-Gaussian model was thus preferred.

The model has a total of 7 parameters for the 16 observations excluding the two zero observations as these cannot be included in the log-Gaussian error model. The zero concentration observations at time zero are however implicitly assumed by the model and it will thus not affect the fit. The model is estimated by defining the likelihood function in R and maximizing it using R's built in optimizer `nlminb`. The parameter estimates are shown in Table 3.2. The blood plasma volume V_p is estimated as 6.3% of the body mass and this corresponds well to an approximate value of 7% that is a commonly used reference value (Lee and Blaufox, 1985). The brain volume V_b is estimated to be much larger than the blood volume and indicates that Gaboxadol is bound in the brain in a form where it is not measured. The smallest rate constant is k_{21} and the release from the brain is thus the rate limiting step.

The fit of the model is shown in Figure 3.3 and in Figure 3.4 the fit is shown on log-scale. In particular from Figure 3.4 it can be noticed how the double exponential decay in the plasma concentration seems to fit well to the observed PK profile. Also, since the release from the brain is the rate limiting step the terminal slopes in both compartments are identical (Gabrielsson and Weiner, 2006). This is illustrated in Figure 3.4 where the brain PK curve is inserted as a dotted line together with the plasma PK curve.

k_a	2.082 min^{-1}
k_e	80.132 min^{-1}
k_{12}	29.292 min^{-1}
k_{21}	0.624 min^{-1}
V_p	0.063 ml/g
V_b	8.943 ml/g
σ	0.255 $\log \mu$g/ml

Table 3.2: Parameter estimates for Gaboxadol PK model.

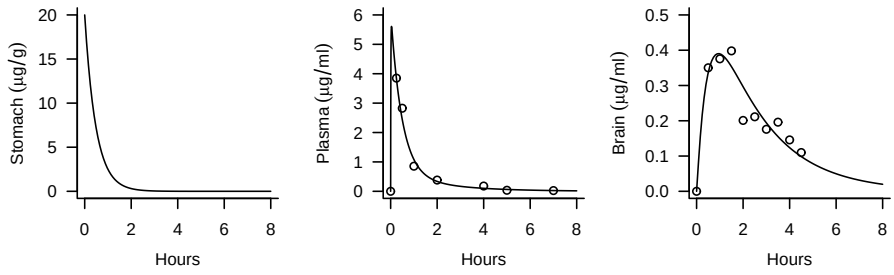


Figure 3.3: PK model for Gaboxadol.

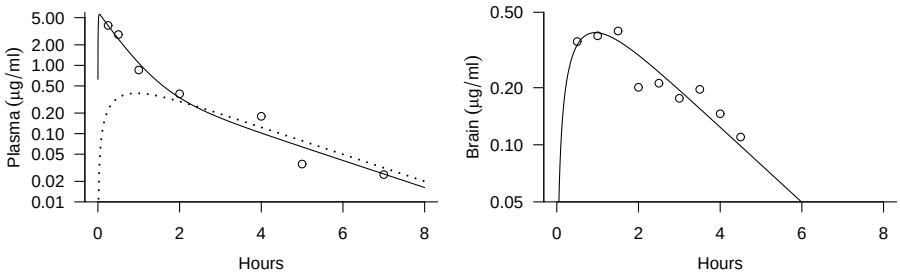


Figure 3.4: PK model for Gaboxadol shown on log-scale. The brain PK curve (right) is inserted as a dotted line together with the plasma PK curve (left).

3.1.3 Discrete versus continuous time

As discussed at the beginning of this chapter the Markov process assumed to generate the observed hypnograms can be modelled in either discrete or continuous time. In this thesis it is chosen to mainly work with the continuous representation for reasons which is argued as follows:

It is not realistic to think that the real sleep process is separated into epochs of an arbitrary length of either e.g. ten or thirty seconds. Sleep is a continuous process and should thus also be modelled as such (Kemp and Kamphuisen, 1986). Of course the scoring of sleep into a number of states is in itself an abstraction and interpretation of the data, but it is appealing to think of the sleep states as a process describing the state of consciousness of the body which may change at any time point and which it is simply chosen to sample at discrete time points.

Moreover, the continuous representation makes the parameterization independent of the sampling period which is not the case for a discrete time representation. This is because the continuous process is parameterized by rate related parameters related to the expected time until next state change whereas the discrete process is defined in terms of probabilities directly related to the sampling interval. For this reason the continuous time parameters may also be easier to relate to without an in-depth understanding of Markov processes.

Finally if the actual process is evolving in continuous time and have constraints on the possible jumps so that not all jumps between all states are allowed this can be directly included in the continuous time model representation. If the model is described in discrete time this may not be the case, since the process may change to any state in a series of jumps and thus such physical restrictions are more difficult or impossible to make use of in a discrete model of the process.

3.2 Model definition

A general description of the continuous time Markov process is given in the following. Let $\{x(t) \in S; t \in T\}$ be a continuous time process with a finite state space $S = \{1, ..., m\}$ and $T = [0, \infty[$. The process is defined by a family of stochastic matrices of transition probabilities $\boldsymbol{P}(t,u) = \{p_{ij}(t,u)\}$, $u > t$, given as

$$p_{ij}(t,u) = \Pr(x(u) = j | x(t) = i). \tag{3.3}$$

Since the transition probabilities are only dependent on the previous state of the process it is by definition a Markov process. To simplify the notation in the following the Markov process is assumed to be homogeneous such that $p_{ij}(t) = \Pr(x(t+u) = j | x(u) = i)$. The functions $p_{ij}(t)$ must satisfy

$$0 \leq p_{ij}(t) \leq 1, \tag{3.4}$$

$$\textstyle\sum_j p_{ij}(t) = 1, \tag{3.5}$$

$$p_{ik}(t) = \textstyle\sum_j p_{ij}(v) p_{ik}(t-v), \quad t > v. \tag{3.6}$$

A process fulfilling (3.5) is called an honest process since it almost surely (i.e. with probability 1) will stay in the state space S. The equation in (3.6) is known as the Chapman-Kolmogorov equation for time homogeneous Markov processes which follows directly from the law of total probability, $\Pr(A) = \sum_j \Pr(B_j) \Pr(A|B_j)$, and the Markov property (Cox and Miller, 1965). The Chapman-Kolmogorov equation in (3.6) makes it possible to build up conditional probabilities over longer time intervals $(0,t)$ from the smaller intervals $(0,v)$ and (v,t). This makes it convenient to define the continuous time Markov process by the transition rates over short time intervals $\Delta t \to 0$. This is done using a first order Taylor expansion of transition probabilities given as

$$\begin{aligned} p_{ij}(\Delta t) &= q_{ji}\Delta t + o(\Delta t), \quad i \neq j, \\ p_{ii}(\Delta t) &= 1 + q_{ii}\Delta t + o(\Delta t). \end{aligned} \tag{3.7}$$

since $p_{ii}(0) = 1$ and $p_{ij}(0) = 0$ and where $o(\Delta t)$ represents a quantity that goes to zero faster than Δt. In this way the time homogeneous continuous Markov process can be defined solely in terms of the matrix of transition rates $\boldsymbol{Q} = \{q_{ij}\}$ also simply know as the $\boldsymbol{Q}$-matrix. Based on (3.7) it is seen that for (3.5) to hold it follows that

$$q_{ii} + \sum_{j \neq i} q_{ij} = 0. \tag{3.8}$$

meaning that the row sum of $\boldsymbol{Q}$ must be 0 and in combination with (3.4) it is also seen that $\boldsymbol{Q}$ has diagonal elements $q_{ii} \leq 0$ and off-diagonal elements $q_{ij} \geq 0$, $i \neq j$. It also holds that $w_{ij} = -q_{ij}/q_{ii}$ is the probability of going to state j when a jump from state i occurs. This can be used to give a different parameterization of $\boldsymbol{Q}$ based on q_{ii} and w_{ij}.

Using (3.7) it is possible to find transition probabilities for very small time steps Δt. To extend this to arbitrary time steps it is necessary to define the forward equations. Suppose that the process starts at state i, $x(0) = i$, and that $p_{ij}(t) = \Pr(x(t) = j|x(0) = i)$. Using (3.6) for $\Delta t > 0$ this gives

$$p_{ik}(t + \Delta t) = p_{ik}(t)(1 + q_{kk}\Delta t) + \sum_{j \neq k} p_{ik}(t) q_{jk} \Delta t + o(\Delta t), \tag{3.9}$$

where the first term is the probability of going directly from i to k and staying there and the last term is the probability of going from i to j and then to k. Letting $\Delta t \to 0$ in (3.9) results in $p'_{ik}(t) = \sum_j p_{ij}(t) q_{jk}$ and in matrix notation this is written as

$$\boldsymbol{P}'(t) = \boldsymbol{P}(t)\boldsymbol{Q} \tag{3.10}$$

with initial condition $\boldsymbol{P}(0) = \boldsymbol{I}$. If a time inhomogeneous Markov process is considered the equation generalizes to

$$\frac{\partial}{\partial u}\boldsymbol{P}(t, u) = \boldsymbol{P}(t, u)\boldsymbol{Q}(u), \tag{3.11}$$

which is known as the Kolmogorov forward differential equation. For a given $\boldsymbol{Q}(t)$ the conditional probabilities governing the process is thus completely described using (3.11). It can be shown that if $\boldsymbol{Q}(t)$ is time invariant then time between jumps (holding times) are exponentially distributed and that the diagonal elements $q_{ii}(t)$ of $\boldsymbol{Q}(t)$ contains the negative rate for leaving a state. This follows from (3.7) since the probability of staying in the same state in any interval Δt is $1 + q_{ii}\Delta t$ which gives a geometric distribution of holding times for $\Delta t > 0$ and an exponential distribution for $\Delta t \to 0$.

3.2.1 Likelihood estimation

The process $x(t)$ is observed at N discrete time points that are chosen independent of the observed process. The model dynamics are assumed to be slowly varying relative to the time between observations. The observation sequence is denoted $\{x_k\}$ and contains the state at time t_k where $k = 1, ..., N$.

The likelihood function is formed as a product of conditional densities that can be found using (3.11) and is given as

$$L(\boldsymbol{Q}(t)) = \prod_{k=1}^{N-1} L_k(\boldsymbol{Q}(t)) \tag{3.12}$$

$$= \prod_{k=1}^{N-1} p(x_{k+1}|x_k) \tag{3.13}$$

This likelihood function is called a conditional likelihood function since it is defined conditional on the first observation.

As noted above, the model dynamics are assumed to be slowly varying relative to the time between observations. This has the implication that $\boldsymbol{Q}(t)$

for $t_k \leq t \leq t_{k+1}$ can be assumed constant between two observations so that $\boldsymbol{Q}_k = \boldsymbol{Q}(t_k)$. With this approximation the transition probabilities in (3.13) can be found using the forward equations in (3.10) as

$$P(t_k, t_{k+1}) = \exp(\Delta t_k \boldsymbol{Q}_k) \tag{3.14}$$

where $\Delta t_k = t_{k+1} - t_k$ and $\exp(\cdot)$ denotes the matrix exponential. The approximation avoids solving the partial differential equations in (3.11). If Δt_k is larger relative to the time variations of $\boldsymbol{Q}(t)$ one can also choose to use a first or higher order expansion of $\boldsymbol{Q}(t)$ which also has an explicit solution for the transition probabilities.

3.2.2 The imbedding problem for Markov chains

For estimation problems where it is assumed that the Markov process evolves in continuous time it could seem tempting to estimate the transition probabilities directly instead of estimating parameters in the continuous time representation where the matrix exponential is involved. However, this leads to problems since the matrix exponential is not a one-to-one transformation and this is related to the imbedding problem for Markov chains.

The imbedding problem for Markov chains is the question about whether a given discrete time Markov chain can be obtained by discrete time sampling of a continuous time Markov jump process. The imbedding problem has received much attention within theoretical analysis of Markov processes going back to Elfving (1937) and later e.g. Kingman (1962) and the problem is also relevant in the present context of estimating parameters using maximum likelihood.

To clarify the problem it is illustrated for a time homogeneous process which is observed with a constant sampling interval Δt (Bladt and Sørensen, 2005). If it is assumed that the observations come from a Markov process in continuous time the parameters $\boldsymbol{Q}$ are estimated using the likelihood function in (3.13). The transition probabilities are constant due to a constant $\boldsymbol{Q}$ and Δt and are estimated as $\exp(\Delta t \hat{\boldsymbol{Q}})$ where $\hat{\boldsymbol{Q}}$ is the MLE of $\boldsymbol{Q}$. If it is instead assumed that the observed sequence of states comes from a Markov process in discrete time, the parameters to be estimated are directly the transition probabilities $\boldsymbol{P}_d$, which define a discrete time Markov process. It can be shown that the maximum likelihood estimate is given as $\hat{\boldsymbol{P}}_d = \{n_{ij}/n_{i\cdot}\}$ where n_{ij} denotes the number of jumps from i to j and $n_{i\cdot}$ the total number of jumps from i.

If there exist a $\boldsymbol{Q}$ fulfilling the criteria for an intensity matrix and such that $\hat{\boldsymbol{P}}_d = \exp(\Delta t \boldsymbol{Q})$ then this $\boldsymbol{Q}$ is the MLE $\hat{\boldsymbol{Q}}$. However, the equation will not always have a solution, in which case the estimated $\hat{\boldsymbol{P}}_d$ does not represent transition probabilities that can be obtained from a continuous time Markov chain. Exactly defining the set matrices $\hat{\boldsymbol{P}}_d$ where the equation can be solved given the constraints on $\boldsymbol{Q}$ is a difficult problem, but a simple and sufficient

criterion is that all diagonal elements of $\hat{\boldsymbol{P}}_d$ are $\geq 1/2$ (Cuthbert, 1973). This indicates that the problem is related to the sampling frequency, since a faster sampling will result in higher diagonal probabilities.

This discussion emphasizes the necessity of estimating parameters in the continuous time representation if this is the model that should be used for inference and in particular if the actual process is known to evolve in continuous time. The continuous time representation involves the extra complexity of using the matrix exponential to evaluate the likelihood function but insures that the estimated parameters will in fact represent a continuous time Markov process.

3.3 Non-parametric estimation

Until now estimation of the time varying $\boldsymbol{Q}(t)$ matrix has been referred to without a specific parameterization in mind. The problem of finding a suitable parameterization is in some sense comparable to the problem of finding a suitable regression function in non-linear least squares regression based on a set of observations. The problem here is that it is not possible to get a visual impression of the time variations of $\boldsymbol{Q}(t)$ by plotting the observed data (see e.g. Figure 3.1) since the rate related parameters of $\boldsymbol{Q}(t)$ cannot be directly related to the individual observations.

To overcome this problem the data can be separated in small time segments where the parameters of the intensity matrix can be estimated by assuming a locally time homogeneous process. This approach is used in Kemp and Kamphuisen (1986) for clinical hypnogram data and in Madsen et al. (1985) for observations of cloud cover and results in rough estimates of the time variations.

In paper D an improved method for estimation of the time variations of $\boldsymbol{Q}(t)$ is presented. The method is based on a locally weighted likelihood function together with a polynomial approximation of the parameters defining $\boldsymbol{Q}(t)$. The method is generally applicable for local estimation in continuous time Markov processes and has a number of advantages in comparison to more simple approaches as the one described above. It is possible to use any choice of kernel for weighing the data and the use of higher order polynomials makes the method more capable of capturing rapid changes in the $\boldsymbol{Q}(t)$ matrix. A typical problem when doing local estimation using e.g. a zero or first order polynomial is that estimates of peaks will be negatively biased since this shape is not well approximated by these lower order polynomials. To avoid this problem it is necessary to use a relatively smaller bandwidth which on the other hand results in a larger variation in the estimates. A second order polynomial is naturally a much better approximation around peaks also for larger bandwidths, which gives more robust estimates since it is possible to use a larger bandwidth without increasing the bias in same way as for the low order polynomial approximations. In paper

D there is a comparison of the results of estimation using 0th, 1st, and 2nd order polynomials where it is seen how the 2nd order approximation is much more sensitive to peaks in the parameters.

The method for local estimation method in paper D uses a set of parameters $\boldsymbol{\beta}$ which defines the local polynomials used to describe the time variations $\boldsymbol{Q}(t)$ around a time point of interest t_c. As in ordinary local estimation methods the idea is to find a local estimate of $\boldsymbol{Q}(t_c)$ by estimating the parameters $\boldsymbol{\beta}$ using locally weighted data. To get a complete picture of the time variations in $\boldsymbol{Q}(t)$ the method is repeated for a series of suitably close values of $t_c \in T$.

The local estimation method for inhomogeneous Markov processes will be outlined in the following in order to provide the basis for discussing extensions of the methods for choosing bandwidth presented in paper D.

3.3.1 Choice of bandwidth

The likelihood function for the local estimation method at a given time point of interest t_c is defined as

$$\log L(\boldsymbol{\beta}, t_c) \quad = \quad \sum_{k=1}^{N-1} w(x_k, t_c) \log L_k(\boldsymbol{\beta}) \tag{3.15}$$

where L_k is the likelihood of a single observed transition defined in (3.13). The weights for the observed transitions are $w(x_k, t_c)$ and are found as

$$w(x_k, t_c) = K_{h(x_k)}(t_k - t_c). \tag{3.16}$$

The kernel function K is a symmetric probability function and $h(x_k)$ is a *state dependent* bandwidth. The bandwidth defines the size of the local neighborhood by scaling the kernel function as $K_h(t) = K(t/h)/h$ (Fan and Gijbels, 1996).

The definition of the weights in (3.16) as state dependent is an extension to the definition in paper D, where the weights are simply defined as $w_k(t_c) = K_h(t_k - t_c)$ independently of the observed state.

The reason for introducing the state dependent bandwidth is that information about the ith row in $\boldsymbol{Q}(t)$ is mainly contained in the observed transitions *from* state i. If there are no observations of transitions from state i it is not possible to estimate any parameters for state i related to holding times or probabilities of jumping to other states. Conversely, if many observations of transitions from state i are available in the data the ith row in $\boldsymbol{Q}(t)$ will be well defined. By using a state dependent bandwidth it is possible to define local bandwidths that include a more even amount of information about the individual states and thereby making the method more equally local for all parameters to be estimated.

Two methods for state dependent bandwidth are considered. The bandwidth for a state i can be chosen such that the interval $t_c \pm h(i)$ contains either

1. a total of M observations of state i, or,

2. a fixed proportion α of the observations of state i (denoted nearest neighbor (NN) bandwidth).

With Method 1 it may happen that M is larger than the observed number of jumps from a state, that is $M > n_{i\cdot}$ for some $i \in S$. To handle this smoothly the bandwidth is increased for these states by a factor $M/n_{i\cdot}$ of the bandwidth covering all observations. In this way the kernel weights given in (3.16) will even out and approach a constant within the observation window for $h \to \infty$. The approximation of parameters with only very limited information will thus tend toward a global polynomial representation.

Methods 1 and 2 differ in that the first method aims at using an equal amount of information to estimate parameters for each state, whereas the second method will use a fixed proportion of the information available for each state. The preferred method will depend on the application at hand.

In paper D an example of a non-parametric estimation of the $\boldsymbol{Q}$ matrix for the EEG hypnogram data from the study of Gaboxadol in rats is illustrated. The estimation is performed by pooling all data from the six rats to give estimates of the mean effects in the data. Only the treatment data is used giving a total of $8,460 \times 6 = 50,760$ observations. The parameterization of $\boldsymbol{Q}$ is given in terms of $q_1(t)$ for the rate for leaving state i and w_i for the probability of jumping to $i-1$ when a transition occurs. The parameter $w_1 = 0$ since physiologically jumps from W to PS should not occur and this can be implemented directly in the continuous time representation. The model is thus defined as

$$\begin{aligned}
\boldsymbol{f} : \boldsymbol{\theta}(t) &\rightarrow \boldsymbol{Q}(t) : \\
\boldsymbol{Q}(t) &= \begin{bmatrix} -q_1(t) & q_1(t) & 0 \\ w_2(t)q_2(t) & -q_2(t) & (1-w_2(t))q_2(t) \\ (1-w_3(t))q_3(t) & w_3(t)q_3(t) & -q_3(t) \end{bmatrix} \\
q_i(t) &= \exp[\theta_i(t)], \quad i = 1,2,3 \\
w_i(t) &= \text{logit}^{-1}[\theta_{i+2}(t)], \quad i = 2,3\,,
\end{aligned} \tag{3.17}$$

where the parameters $\boldsymbol{\theta}(t) = [\theta_1(t), ..., \theta_5(t)]$ are estimated locally using 2nd order polynomials defined by the parameters in $\boldsymbol{\beta}$. With 5 parameters in the model and using 2nd order polynomials this gives a total of 15 parameters in $\boldsymbol{\beta}$ that is estimated for every time point t_c. When defining the parameterization of the model it is necessary to make the model unbounded in the $\boldsymbol{\theta}(t)$ parameters which is done here using the exponential and logit transform for the rate and probability parameters respectively. If the model is not unbounded in $\boldsymbol{\theta}(t)$ the polynomial representation of these parameters may easily give values of the parameters where the model cannot be evaluated to find the transition probabilities.

In paper D the parameters $[\theta_1(t), ..., \theta_5(t)]$ are estimated locally using 2nd order polynomials and a state *in*dependent NN bandwidth of $\alpha = 0.40$ of the total number of observations. The result is seen in Figure 3.5. For comparison, the result of estimation using the state dependent bandwidth in Method 1 with $M = 6000$ and using 2nd order polynomials is shown in Figure 3.6 with the bandwidth for the individual states shown in the last row. For both figures a tricube kernel function has been used. For the NN method using $\alpha = 0.40$ the bandwidth is constantly $0.40 \times 23.5\,\text{h.} = 9.4\,\text{hours}$ since the sampling times are equidistant For the state dependent bandwidth the bandwidth varies since the frequency of visits to the three states differ throughout the time period. In particular it is seen that the bandwidth is as low as 3-4 hours for the DS state during the initial 12 hours and using the 2nd order polynomial this gives a much more apparent effect peak in expected time in the DS state with a maximum of more than 12 minutes compared to 6 minutes for NN method.

For the PS state there is only a total of 1343 observations ($\ll M$) giving a constant bandwidth of $23.5\,\text{h} \times 6000/1343 = 105\,\text{hours}$. This results in tricube weights between 1 and 0.9955 for a 24 hour range which again results in almost global polynomial approximations for the two parameters related to the PS state. The global estimates for the two parameters q_3 and w_3 for the PS state are thus similar to 2nd order polynomials that has been either log or inverse-logit transformed but deviations are still seen since the estimates are correlated with the other more locally estimated parameters.

3.3.1.1 Relation of effect to the PK-profile

To compare the effect of Gaboxadol on the non-parametric estimates of the sleep parameters the analysis has been carried out on both placebo and treatment data. The results are shown in Figure 3.7 using $M = 6000$, the same as used for Figure 3.6. It is seen that the most apparent effect of Gaboxadol is found in the estimates of the expected time in the DS state. The significant peak between 0 and 6 hours is only found in the treatment data, whereas the placebo data seems to be rather constant during this period. The effect on the expected time in the DS state can thus be attributed to Gaboxadol, and it is thereforeinteresting to see how the estimated effect relates to the mean PK concentration profile of Gaboxadol in the brain which is modelled in Section 3.1.2.

In Figure 3.8 the expected time in the sleep state DS is compared with the PK profile of Gaboxadol in the brain. The figure indicates that there is a direct relation between the PK profile and the expected time in the DS state with a delay of approximately one hour. The indication of this relation directly benefits from the high temporal resolution and sensitivity of the estimate of the expected time in DS found using the non-parametric local estimation method.

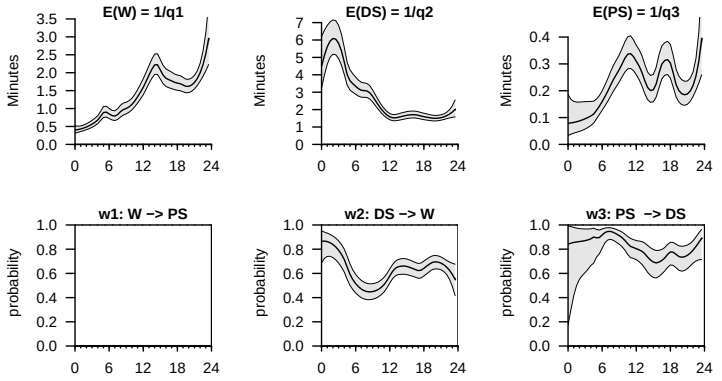


Figure 3.5: Estimate of the $\boldsymbol{Q}(t)$ matrix as a function of time using 2nd order local polynomials and a NN bandwidth $\alpha = 0.40$. The surrounding lines are Wald 95% pointwise confidence intervals.

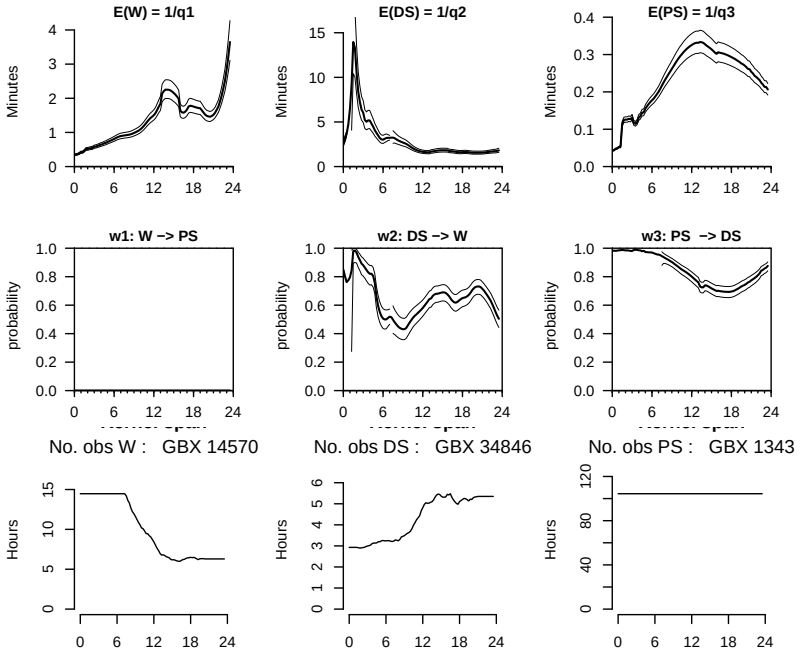


Figure 3.6: Estimate of the $\boldsymbol{Q}(t)$ matrix as a function of time using 2nd order local polynomials and a state dependent bandwidth $M = 6000$. Plots in the 3rd row show bandwidths for (and number of jumps from) the individual states.

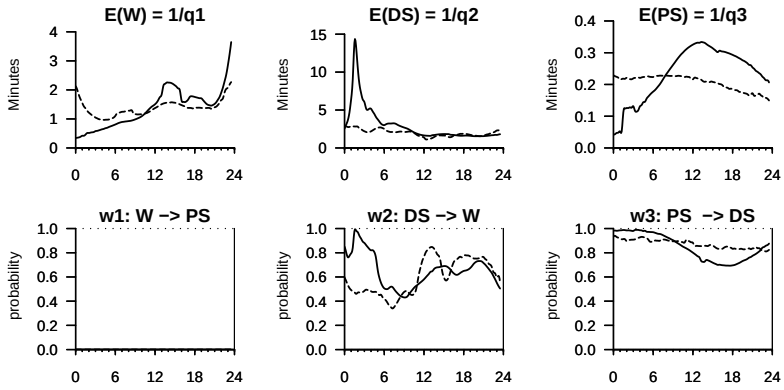


Figure 3.7: Estimate of the $\boldsymbol{Q}(t)$ as in Figure 3.6. The Gaboxadol treatment is shown in a solid line and placebo in a dashed line.

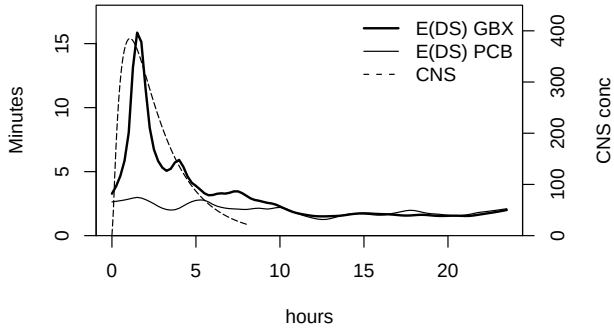


Figure 3.8: Expected time in DS state versus brain PK-profile.

3.3.2 Numerical issues

Depending on the application and amount of data, the numerical implementation of the local estimation method for Markov processes requires some consideration to give reasonable estimation times. In the example using the sleep states from rats there are 8,460 observations for each rat and treatment. It is necessary to solve the Kolmogorov forward differential equation for each observation to find the probability of the transition given $\boldsymbol{\beta}$. Since the sampling period is only 10 seconds, which is very short relative to the time variations of the parameters, this is done using the matrix exponential by assuming constant parameters over the sample intervals, see (3.14). When estimating using the pooled data the matrix of transitions probabilities for all possible jumps for a single time point can be reused for all six rats, but this still means that the matrix exponential

is evaluated 8,460 times for each evaluation of the likelihood function.

The implementation of the local estimation method has been done mainly in R. The default R installation does not have a matrix exponential function so the initial work was done using the Matrix package that uses a method by Ward (1977) based on a Pade' approximation with three preconditioning steps which has originally been implemented for Octave (Eaton, 2002).

A small side result from this initial work with the Matrix package was the discovery of an error in the code which resulted in erroneous evaluations of the matrix exponential that was evident for certain matrices. This was pointed out to Martin Maechler, a member of the R Development Core Team, who was able to locate the bug which turned out to come from the original Octave code.

The final implementation of the estimation method is partly based on R with the evaluation of the likelihood function (3.15) including the matrix exponential being done in Fortran. The matrix exponential is evaluated using `DGPADM` from EXPOKIT (Sidje, 1998) which uses a Pade approximation to the exponential function combined with scaling-and-squaring (Moler and Van Loan, 1978). By keeping the likelihood function solely in Fortran gives a significant speedup. The optimization of the parameters in the likelihood function is done by calling general optimizers from R. This was found as a good compromise between coding complexity and flexibility.

The optimization problem is in itself relatively complex. As mentioned there are a total of 15 parameters in $\boldsymbol{\beta}$ in the model for the rat EEG data that is estimated for every time point t_c and these parameters are all dependent on each other. The optimization problem at each t_c benefit from the fact that they are serially related such that the final estimate of $\boldsymbol{\beta}$ at t_c is a good initial value at the next $t_c + \Delta t$. The estimation of $\boldsymbol{\beta}$ has been done with a standard quasi-Newton method with BFGS updating of the Hessian for finding the search direction and using soft line search for finding the next iterate (Nocedal and Wright, 2006). This method generally performs well for optimization of likelihood functions and is available in `optim` in the stats package in R. The method initializes the Hessian as the identity matrix which is also most commonly done. However, this can be improved in the serial optimization structure by using the final Hessian estimate from the previous t_c as the initial estimate in the next optimization. This feature is available for the ucminf method (Nielsen, 2000) and was implemented in R for this purpose (Nielsen and Mortensen, 2008). The method works well but it is not found that the improved initial estimate of the Hessian gives a notable reduction in estimation times. The `ucminf` method were afterwards also used for the estimation problems with mixed models in paper E and here it is in fact able to outperform `optim` and `nlminb` both from the stats package in R.

3.4 Parametric estimation

The non-parametric estimates of the time variations of the Markov process can be used to identify a possibly more simple parametric description of these time variations. This can be used for modelling the main structures in the parameters by testing for model reduction using a standard likelihood ratio test (Madsen, 2008). Also in a parametric model individual variations can be described using a mixed model with random effects and the drug effect can be included as function of the PK concentration profile, which gives the possibility of statistically testing different relations with PK profile for the parameters in the Markov process.

It is noted that the mixed modelling approach modifies the Markov assumption to be conditional on the random effects. When the random effects are integrated out, the process is nolonger of the Markov type. One can see this as a more flexible class of processes, which can still be interpreded in a Markov framework.

Karlsson et al. (2000) use the parametric approach to describe the sleep structure in a study of 21 patients that are treated with both placebo and the sleep drug Temazepam. The modelling is not done in continuous time but instead in discrete time where each transition probability is modelled separately. This means that for modelling jumps e.g. from state 1 to 2 all data with jumps from state 1 is extracted and it is noted if a jump to state 2 occurs or not. The data for a particular transition from i to j is thus binary $y_k \in \{0, 1\}$, $k = 1, ..., n_{i\cdot}$, and the modelling of each transition probability as a function of time is similar to a logistic regression problem. The main focus of the modelling is to incorporate the drug effect in the model and also to include the individual variation using Gaussianly distributed random effects. Both the drug effect and random effects are included additively on the logit scale of the transition probability which ensures that probabilities are always in the interval $[0; 1]$. The model is fitted using NONMEM to handle the estimation of the mixed model. NONMEM has the possibility of directly specifying the log-likelihood for the first stage model which is given as $\log L_1(\boldsymbol{b}, \boldsymbol{\theta}) = y_k p^*_{ij}(\boldsymbol{b}, \boldsymbol{\theta}, t) + (1-y_k)(1-p^*_{ij}(\boldsymbol{b}, \boldsymbol{\theta}, t))$ where $p^*_{ij}(\boldsymbol{b}, \boldsymbol{\theta}, t)$ denotes the transition probability as a function of the random effects $\boldsymbol{b}$ and parameters $\boldsymbol{\theta}$ in the model. The model is fitted using the 'Laplace' option in NONMEM which uses the likelihood in (2.12).

The approach in Karlsson et al. (2000) has also been applied to the rat EEG data to investigate the individual difference between the rats. The transition from DS to W is considered and rough estimates of the individual transition probabilities are shown in Figure 3.9. The estimates are found by counting the number of transitions from DS to W in two hour intervals and dividing by the number of observations of DS in this interval.

The transition probability is modelled with a simple model with two levels

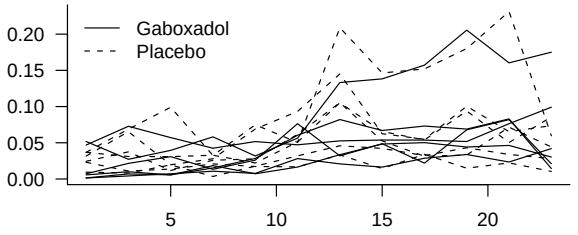


Figure 3.9: Individual probabilities of going from DS to W.

of the transition probability θ_1 and θ_2 given as

$$f(t, \boldsymbol{\theta}) = \frac{\exp(t - \theta_3)^{\theta_4}(\theta_2 - \theta_1)}{(1 + \exp(t - \theta_3)^{\theta_4})} + \theta_1 \tag{3.18}$$

where t is the point in time, θ_3 is the time of the change from θ_1 to θ_2 and θ_4 defines the smoothness of this change such that for $\theta_4 = \infty$ the model is a step function with two levels θ_1 and θ_2. Similarly to Karlsson et al. (2000) the drug and individual random effect is included as

$$\operatorname{logit} p^*_{ij}(t_k) = \operatorname{logit}[f(t, \boldsymbol{\theta})] + \theta_5 c(t_k) + b_{i'} \tag{3.19}$$

where $c(t)$ is the concentration of Gaboxadol in the brain at time t ($c(t) = 0$ when the rat receives placebo) and $b_{i'} \sim N(0, \sigma^2)$ is the random effect for rat i'. The model for the transition probability is shown in Figure 3.10 for a number of draws of random effect to illustrate how they affect the model. The model

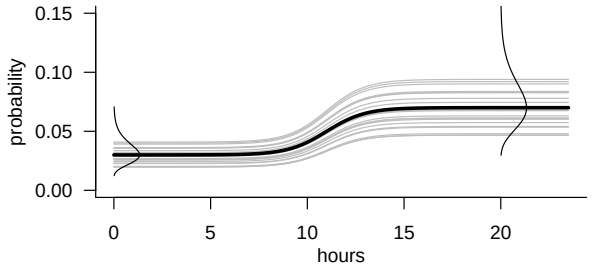


Figure 3.10: Model for transition probability from DS to W.

is fitted using NONMEM. To focus on the individual differences two random effects are estimated for each rat for the placebo and Gaboxadol treatment. The estimates of the random effects are shown in Figure 3.11 where the two random effects for each rat is plotted against each other together with a $y = x$ line. From Figure 3.11 it is seen that the estimates of the random effects are almost identical

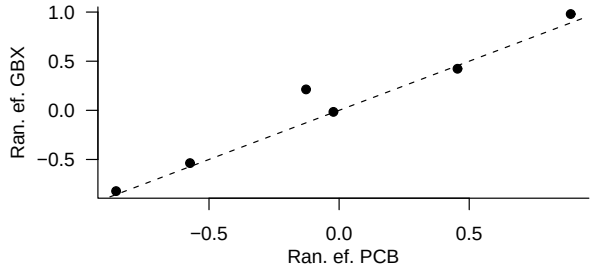


Figure 3.11: Model for transition probability from DS to W.

between the two nights. This is interesting since it indicates that there is an inherent sleep pattern for each rat which does not vary between the two nights where they are observed. Some rats simply seem to have a higher tendency to go from DS to W than others. This results in a more rapidly oscillating sleep pattern in the hypnogram which is found to correspond well to estimates of the random effects for each rat.

The parametric description of the sleep structure can also be done in continuous time by defining the $\boldsymbol{Q}(t)$ matrix as a function of a set of parameters $\boldsymbol{\theta}$ that are estimated using (3.12). Similarly to the discrete time parametric estimation it is also possible to include random effects in such a model to account for individual differences by using the Laplace likelihood for mixed models given in (2.12). However, since the entire matrix $\boldsymbol{Q}(t)$ must be modelled and estimated in a single model, such a model very easily becomes very complex if it should adequately describe the time variations in all parameters that are found using non-parametric estimates. For this reason a parametric description of $\boldsymbol{Q}(t)$ for the rat EEG data has not been pursued any further.

Given a parameterization of $\boldsymbol{Q}(t)$ for a continuous time model that involves both fixed and random effects, the maximum likelihood estimation of such a model is a challenge in itself as it involves both solving the Kolmogorov forward differential equation and using the Laplace likelihood. If such a model were to be applied it is necessary perform the estimation in a computationally efficient frame work to give reasonable estimation times, and it will thus be outlined how the estimation can be done in NONMEM.

NONMEM does not include a matrix exponential function, but instead it is very efficient for solving differential equations as these are used in compartmental PK modelling. This can be used to find the conditional probabilities of the observations by solving the forward equation (3.10) directly with the initial condition given by the previous observation. In a system with three states there

are three differential equations to solve for $\boldsymbol{p}_i = [p_{i1}\ p_{i2}\ p_{i3}]$ given as

$$\frac{d}{dt}\boldsymbol{p}_i = \boldsymbol{p}_i \begin{bmatrix} q_1 & q_{12} & q_{13} \\ q_{21} & q_2 & q_{23} \\ q_{31} & q_{32} & q_3 \end{bmatrix} \tag{3.20}$$

but since the conditional probabilities must sum to 1, it is only necessary to solve two of the differential equations to find the probabilities p_{i1} and p_{i2}. Depending on the previously observed state the initial condition for $[p_{i1}\ p_{i2}]$ will be either [1 0], [0 1] or [0 0] for state 1, 2, and 3, respectively. The forward differential equations for p_{i1} and p_{i2} becomes

$$\frac{d}{dt}\begin{bmatrix} p_{i1} \\ p_{i2} \end{bmatrix} = \begin{bmatrix} q_1 - q_{31} & q_{21} - q_{31} \\ q_{12} - q_{32} & q_2 - q_{32} \end{bmatrix}\begin{bmatrix} p_{i1} \\ p_{i2} \end{bmatrix} + \begin{bmatrix} q_{31} \\ q_{32} \end{bmatrix} \tag{3.21}$$

and they can be derived in a similar way for a system of any size. This system can be solved in NONMEM by considering p_{i1} and p_{i2} as two pseudo compartments which are reset before each observation to [0 0] using a reset event in the data file and possibly updated to [1 0] or [0 1] with an additional dose event if the previous observation is either 1 or 2, respectively. When the differential equations are solved for each observation to give the conditional probabilities of the observation, the first stage likelihood can be formed in NONMEM in the same way as it is done for the discrete time Markov models as discussed previously. This approach has been implemented in NONMEM and was found to estimate the parameters in the $\boldsymbol{Q}(t)$ matrix correctly by testing using simple parametric models. The estimation method in NONMEM opens up for experimentation with the parametric approach for continuous Markov processes in combination with mixed effects modelling, but as mentioned previously this has not been pursued further here.

3.5 Discussion

Although local estimation of parameters in an inhomogeneous Markov process is a promising way of extracting information about the sleep process, it should still be kept in mind that any results and interpretations are limited by the model used for scoring sleep stages.

For human sleep staging the guidelines by Rechtschaffen and Kales has been strongly debated over the past many years (Himanen and Hasan, 2000). The guidelines were developed for healthy adults with undisturbed sleep and the scoring may thus be ambiguous for patients with sleep disturbances. The guideline in general leaves some room for interpretation and this results in inter-rater agreements around 90% when raters are compared. This observation suggests using a hidden Markov model by assuming that stages are observed with measurement error.

Such a model of observing states with 'noise' (misclassification) is equivalent to the model with SDEs observed with measurement error as presented in Chapter 4. The process defined by SDEs is also Markov but evolves in a continuous space where as the sleep process only evolves in a small discrete set of sleep stages.

A number of replacements to the Rechtschaffen and Kales guidelines has been suggested such as Himanen et al. (1999) and Grube et al. (2002). However, none of these have gained wide spread usage and the original guidelines are thus still used as the gold standard. The most likely reason for this is that although new methods may deal with some of the shortcomings in the original guidelines it is still necessary to use them to be able to compare with past results in sleep studies. For this reason it is very relevant to consider improved analysis of the original sleep stages by using new methods such as the non-parametric approach presented here. This will help increase the understanding of the underlying sleep process while still being kept in a framework that can be related to other results in the area.

Chapter 4

Stochastic differential equations

Stochastic differential equations (SDEs) represent an extensively studied mathematical theory that has found a wide range of practical applications. SDEs provide the tool for building dynamical models based on differential equations where it is possible to include some amount of randomness or noise into the model. A very simple example of such a model is an exponential population growth model where the growth rate is only approximately known and thus loosely given by $r(t)+"noise"$. The model for the population growth can therefore be written as $da(t)/dt = (r(t) + "noise")a(t)$.

To explain this in popular terms, consider a constant growth rate model, say with a growth of 3%/year. An ordinary differential equation model will convert this to an exponential formula for the population. If, for some reason, the growth is higher in one year the model implies a reduced growth growth the following year so the population returns to the originally projected curve. An SDE model will instead accept the higher growth rate in that year and use the actual population as the base for future growth, applying and expected growth of 3%/year.

In order to understand how this can be described mathematically it is necessary to look more closely at the theory defining SDEs.

4.1 Brownian motion

Brownian motion is fundamental to the interpretation of stochastic differential equations as it shows up as the continuous time version of a discrete random walk. The random walk process is given as

$$x_n = z_1 + z_2 + ... + z_n \tag{4.1}$$

where $\Pr(z_i = -1) = \Pr(z_i = 1) = 1/2$. The process in (4.1) is a discrete time process on the integer number line which will move one up or down at each step. Based on z_i define a continuous time process

$$y_t = \sqrt{\Delta t}(z_1 + z_2 + ... + z_{[t/\Delta t]}) \tag{4.2}$$

where $[t/\Delta t]$ is the integer part of $t/\Delta t$, then for $\Delta t \to 0$ it holds that $\mathsf{E}[y_t] = 0$ and $\mathsf{V}[y_t] = \sqrt{\Delta t}^2[t/\Delta t]\mathsf{V}[z_i] \to t$. The central limit theorem implies that $y_t \to \beta_t$ where β_t is Gaussian $N(0, t)$.

It can be shown that the limiting process β_t is in fact standard Brownian motion (Gard, 1988), which is a special form of the more general Wiener process which does not necessarily have mean zero. As can be seen from above it holds for Brownian motion that $\Pr(\beta_0 = 0) = 1$ and that it has stationary and independent increments. This means that for any $t > s$ and $h > 0$ the distribution of $\beta_{t+h} - \beta_{s+h}$ is the same as the distribution of $\beta_t - \beta_s$ (namely $N(0, t-s)$) and for non-overlapping time intervals $[t_1, t_2]$ and $[t_3, t_4]$, the random variables $\beta_{t_2} - \beta_{t_1}$ and $\beta_{t_4} - \beta_{t_3}$ are independent. It can also be shown that β_t is continuous with probability one (Øksendal, 2007), and it will be assumed in the following that β_t is one such version.

4.2 Itô integrals

In a more general form a dynamical model based on differential equations including noise as discussed in the beginning may be written as

$$\frac{dx_t}{dt} = b(t, x_t) + \sigma(t, x_t)w_t \tag{4.3}$$

where x_t is a stochastic process defined as $\{x(t, \omega) \in \mathbb{R} \,|\, \omega \in \Omega, t \in T\}$ where usually $T = [0, \infty[$ and Ω is the ensemble of the process with all possible outcomes (Madsen, 2008). For fixed t, $x(t, \cdot)$ is a random variable and for fixed ω, $x(\cdot, \omega)$ is a realization of the process. The first term in (4.3) is called the drift and the second term the diffusion term. The noise in the system in (4.3) is represented by the stochastic process w_t which drives the process. It is not directly clear how the differential equation in (4.3) defines the stochastic process x_t or which process is the most appropriate to use for w_t. From an application

point of view the process w_t should at least approximately have the following properties (Øksendal, 2007)

1. $t_1 \neq t_2 \implies w_{t_1}$ and w_{t_2} are independent,
2. w_t is stationary,
3. $\mathsf{E}[w_t] = 0$ for all t.

These properties describe a continuous *white noise* process. However, true continuous white noise is a mathematical abstraction since such a process would have a constant power spectral density function for all frequencies (Jazwinski, 1970). This requires the process to have infinite power and it is thus not physically realizable. Hence there is no "reasonable" stochastic process with a continuous sample path satisfying both 1 and 2. If instead a discrete version of (4.3) is considered this will look like

$$x_{k+1} - x_k = b(t_k, x_k)\Delta t_k + \sigma(t_k, x_k)w_k\Delta t_k. \tag{4.4}$$

If $w_k\Delta t_k$ is replaced by $\Delta V_k = V_{t_{k+1}} - V_{t_k}$ then it is seen based on 1 to 3 above that V_t should have stationary and independent increments with mean zero. It turns out that the only process with a continuous path fulfilling this is the Brownian motion. By setting $w_k\Delta t_k = \Delta\beta_k$ in (4.4) gives a corresponding stochastic differential equation of the form

$$dx_t = b(t, x_t)dt + \sigma(t, x_t)d\beta_t. \tag{4.5}$$

The white noise process is thus replaced by the infinitesimal increments of Brownian motion. Although Brownian motion is nowhere differentiable (with probability 1) it can be shown that white noise is the *formal* derivative of Brownian motion $w_t \sim d\beta_t/dt$ (Jazwinski, 1970), which is another argument for choosing the construction in (4.5). If standard integration notation is used to solve (4.5) it results in the equation

$$x_t = x_0 + \int_0^t b(s, x_s)ds + \text{“}\int_0^t \sigma(s, x_s)d\beta_s\text{”}. \tag{4.6}$$

This gives a definition of the stochastic process x_t in (4.5) as a process that satisfies (4.6). What is left is to look more closely at how the stochastic integral

$$\text{“}\int_0^t \sigma(s, x_s)d\beta_s\text{”} \tag{4.7}$$

should be interpreted. In ordinary calculus the integral is defined as the limit of an infinite sum, which in the Riemann-Stieltjes form is given as

$$\int_0^t f(t)dg(t) = \lim_{n\to\infty} \sum_{j=1}^{\infty} f(\tau_j)(g(t_{j+1}) - g(t_j)) \tag{4.8}$$

where τ_j is in the interval $[t_j, t_{j+1}]$. For a deterministic function g under certain regularity conditions this limit converges to a unique value independent of how τ_j is chosen in the interval $[t_j, t_{j+1}]$. The stochastic integral in (4.7) can be defined in the same way by replacing $g(t)$ with the sample path of Brownian motion β_t. Unfortunately the sample path β_t is not sufficiently smooth to define it in the Riemann-Stieltjes sense since β_t has independent increments and therefore is almost nowhere differentiable and has unbounded variance (Øksendal, 2007). Instead, it turns out that the expectation of the stochastic integral in (4.8) with $g(t) = \beta_t$ depends on how τ_j is chosen in the interval $[t_j, t_{j+1}]$. The following choices have proven to be the two most useful

- $\tau_j = t_j$ which defines the Itô integral, or,
- $\tau_j = (t_j + t_{j+1})/2$ which defines the Stratonovich integral.

The choice of which interpretation is used depends on the particular application of the model, but it is important to note that the stochastic process x_t defined as an SDE as in (4.5) can only be understood through the choice of interpretation of the stochastic integral. In biological systems the noise process $d\beta_t$ is often thought to represent discrete pulses with finite separation to which the system responds and for this the Itô interpretation is the most appropriate (Turelli, 1977). In these type of applications the Itô interpretation has thus been the most widely applied and it is also the interpretation that will be used in the following.

When the Itô interpretation is chosen it can be shown that the resulting process x_t is a Markov process: the future development of the process from time t depends only on x_t and not on any previous history of the process. This can be shown by considering (4.5) based on small time increments δt

$$x_{t+\delta t} - x_t = b(t, x_t)\delta t + \sigma(t, x_t)(\beta_{t+\delta t} - \beta_t) \tag{4.9}$$

where b and σ are evaluated at x_t due to the Itô interpretation (Jazwinski, 1970). For a given x_t then $x_{t+\delta t}$ will only depend on the Brownian motion increment $\beta_{t+\delta t} - \beta_t$ which is independent and specifically independent of x_t. Therefore the distribution of $x_{t+\delta t}$ depends only on x_t and the process is thus a Markov process. The stochastic process defined with the Itô interpretation is thereby a generalization of a continuous time Markov process in discrete state space discussed in Chapter 3 to a Markov process with a continuous state space.

4.3 Filtering problem

Modelling using SDEs makes it possible to define dynamical models that includes randomness in the system. However, in most practical applications it is not

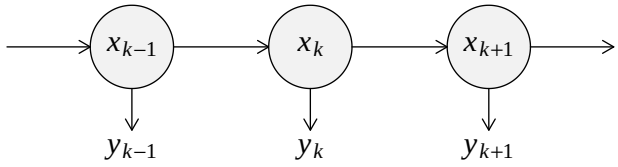


Figure 4.1: Hidden Markov model with states x_k and observations y_k.

possible to observe the state of the system directly without error and this results in a hidden Markov model as illustrated in Figure 4.1.

Estimation of the outcome of a stochastic process defined by an SDE model based on noisy observations is called the filtering problem. The problem will be discussed here for both a univariate stochastic process and measurement but may readily be generalized to higher dimensions for both.

The stochastic process is given as in (4.5) and defines the evolution of the system and is therefore denoted the system equation where x_t is the state of the system. The system is observed at discrete time points t_k with observations given as

$$y_k = h(x_k, t_k) + e_k \tag{4.10}$$

which is denoted the measurement equation where $e_k \sim N(0, S)$ and e_k is serially independent and independent of β_t. If the set of observations until time t_k is defined as $\mathcal{Y}_k = \{y_1, y_2, ..., y_k\}$ the filtering problem is more precisely defined as finding the probability distribution of the state at time t given $\mathcal{Y}_k$, that is finding

$$p(x, t|\mathcal{Y}_k)$$

for $t \geq t_k$. Since the process x_t is a Markov process the evolution of the probability distribution can be described as the conditional distribution given an initial condition $p(x, t|\mathcal{Y}_k)$ where

$$\frac{\partial p(x,t|\mathcal{Y}_k)}{\partial t} = -\frac{\partial p(x,t|\mathcal{Y}_k)b(x,t)}{\partial x} + \frac{1}{2}\frac{\partial^2[p(x,t|\mathcal{Y}_k)\sigma^2(x,t)]}{\partial x^2} \tag{4.11}$$

which is known as the Kolmogorov forward equation or the Fokker-Planck equation (Jazwinski, 1970).

What remains to be shown is how to update the probability distribution $p(x, t_k|\mathcal{Y}_{k-1})$ to $p(x, t_k|\mathcal{Y}_k)$ when a new observation y_k is obtained. By noting that $p(x, t_k|\mathcal{Y}_k) = p(x, t_k|y_k, \mathcal{Y}_{k-1})$ and using Bayes' theorem gives

$$p(x, t_k|\mathcal{Y}_k) = \frac{p(y_k|x_k, \mathcal{Y}_{k-1})p(x, t_k|\mathcal{Y}_{k-1})}{p(y_k|\mathcal{Y}_{k-1})}. \tag{4.12}$$

Since residuals are independent $p(y_k|x_k, \mathcal{Y}_{k-1}) = p(y_k|x_k) = p(e_k|x_k)$ which is directly given by the Gaussian probability density function. The term $p(x, t_k|\mathcal{Y}_{k-1})$

can be found using (4.11) and also it holds that

$$p(y_k, \mathcal{Y}_{k-1}) = \int p(y_k|x)p(x, t_k|\mathcal{Y}_{k-1})dx.$$

Using (4.11) and (4.12) together gives an iterative filter with a prediction equation to find $p(x, t_k|\mathcal{Y}_{k-1})$ and an updating equation to find $p(x, t_k|\mathcal{Y}_k)$. This filter can be used iteratively to find the conditional distribution of the state given the available observations.

4.3.1 Conditional moments

Although the filter described above fundamentally solves the filtering problem, the partial differential equation in (4.11) is not easy to work with. Instead it is a common approach in filtering theory to describe conditional distributions only by their means and covariances which is known as a second order filter. If $\boldsymbol{x}$ denotes a multi-dimensional state, then $p(\boldsymbol{x}, t_k|\mathcal{Y}_{k-1})$ can be described by

$$\hat{\boldsymbol{x}}_{t|t_{k-1}} = \mathsf{E}[\boldsymbol{x}_t|\mathcal{Y}_{k-1}] \tag{4.13}$$

$$\hat{\boldsymbol{P}}_{t|t_{k-1}} = \mathsf{E}[(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|_{k-1}})(\boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|_{k-1}})^T|\mathcal{Y}_{k-1}] \tag{4.14}$$

The two moments of the conditional distribution of the state given $\mathcal{Y}_{k-1}$ at time t_k is thus denoted as $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{P}}_{k|k-1}$. Similarly using the Bayes' theorem for updating leads to $\hat{\boldsymbol{x}}_{k|k}$ and $\hat{\boldsymbol{P}}_{k|k}$ given $\mathcal{Y}_k$.

Estimating the two moments forward in time will give a Gaussian approximation to the conditional distributions, which in the general case is only an approximation. A special case arises for the linear model when the three functions b, σ, and h defining the system and observation equations in (4.5) and (4.10) are linear in the state and the σ function is further independent of $\boldsymbol{x}_t$. For this case it can be shown that if the initial distribution $p(\boldsymbol{x}, t_1)$ is Gaussian then all following conditional distributions are also Gaussian and thus completely described by their mean and variance (Jazwinski, 1970). In this case it is possible to derive explicit solutions for the conditional moments $\hat{\boldsymbol{x}}_{k|k-1}$, $\hat{\boldsymbol{P}}_{k|k-1}$, $\hat{\boldsymbol{x}}_{k|k}$ and $\hat{\boldsymbol{P}}_{k|k}$ and the resulting filter is known as the Kalman filter (KF) (Kalman, 1960). The Kalman filter is described in algorithmic form in paper C.

Since the conditional distributions are described completely by their mean and covariance in the linear case the filter state is said to be 2 dimensional (Jazwinski, 1970). For non-linear models on the other hand the filter state is infinite dimensional as it cannot be described by any finite number of parameters; it is represented exactly only through the complete density $p(\boldsymbol{x}, t|\mathcal{Y}_k)$. However, if it is assumed that the sampling interval is 'small' relative to the degree of non-linearities in the system it is reasonable to assume that the conditional

probabilities are *approximately Gaussian* and thus well approximated by their two first moments. This is due to the fact that both the Brownian motion increments are Gaussian and also the measurement error is Gaussian. These assumptions leads to the Extended Kalman filter (EKF) which handles non-linear models by repeatedly linearizing the model and estimating the two first moments of the conditional densities. For a description of the EKF used for estimation of embedded parameters in an SDE based model see Kristensen and Madsen (2003).

4.4 Likelihood estimation

In statistical applications of models based on SDEs it is necessary to be able to estimate a set of parameters $\boldsymbol{\theta}$ which describes the model. With a slight change of notation from the previous the model can be written as

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t, \boldsymbol{\theta})dt + \boldsymbol{\sigma}_\omega(t, \boldsymbol{\theta})d\boldsymbol{\omega}_t \tag{4.15}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, t_k, \boldsymbol{\theta}) + \boldsymbol{e}_k \tag{4.16}$$

where $\boldsymbol{\omega}_t$ is standard Brownian motion, $\boldsymbol{e}_k \sim N(\boldsymbol{0}, \boldsymbol{S})$ is white noise and $\boldsymbol{e}_k$ and $\boldsymbol{\omega}_t$ are independent and both the state $\boldsymbol{x}_t$ and observations $\boldsymbol{y}_k$ are allowed to be multi-dimensional. The combination of (4.15) and (4.16) is called a (stochastic) state space model. If the system was observed directly (i.e. without noise in the measurement equation (4.16)) the likelihood function could be formed as a product of the conditional probabilities of the observed states in the same way as it is done in Chapter 3 in (3.13) for the Markov model with a discrete state space. However, the model here is a hidden Markov model where the states are not observed directly so instead the likelihood function is formed as a product of conditional densities of the observations $\boldsymbol{y}_k$ based on their one-step predictions. Given the second order moment representation for the state as $\hat{\boldsymbol{x}}_{k|k-1}$ and $\hat{\boldsymbol{P}}_{k|k-1}$ then the moment representation of the conditional distribution of the next observation $p(\boldsymbol{y}_k|\mathcal{Y}_{k-1})$ is found as

$$\hat{\boldsymbol{y}}_{k|k-1} = \mathsf{E}[\boldsymbol{y}_k|\mathcal{Y}_{k-1}] = \boldsymbol{h}(\hat{\boldsymbol{x}}_{k|k-1}, t_k, \boldsymbol{\theta}) \tag{4.17}$$

$$\hat{\boldsymbol{R}}_{k|k-1} = \boldsymbol{C}\hat{\boldsymbol{P}}_{k|k-1}\boldsymbol{C}^T + \boldsymbol{S} \tag{4.18}$$

where $\boldsymbol{R}_{k|k-1}$ is found using the law of error propagation with

$$\boldsymbol{C} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}_t}\Big|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_{k|k-1}, t=t_k, \boldsymbol{\theta}}. \tag{4.19}$$

This approximation is good when the measurement function is approximately linear in the close neighborhood of $\hat{\boldsymbol{x}}_{k|k-1}$ and this is usually the case. Using

(4.17) and (4.18) the likelihood function for $\boldsymbol{\theta}$ is found as

$$L(\boldsymbol{\theta}|\mathcal{Y}_N) = p(\boldsymbol{y}_1|\boldsymbol{\theta}) \prod_{k=2}^{N} p(\boldsymbol{y}_k|\mathcal{Y}_{k-1}, \boldsymbol{\theta}). \tag{4.20}$$

The probability of the first observation in found given an initial Gaussian distribution of the state which may fixed or estimated by including it as a part of the likelihood function for $\boldsymbol{\theta}$.

4.5 Mixed models with SDEs

Modelling using stochastic differential equations is a strong tool for biological systems such as those encountered in PK/PD modelling. PK/PD models are usually based on ordinary differential equations (ODEs) which assumes an entirely deterministic model for the biological system. SDEs are a natural extension to this since they are designed to model partly unpredictable fluctuations in a system. Although indeed actual random fluctuations will often exist and have an important influence a biological system, the fluctuations modelled by SDEs do not necessarily have to be truely random but can also be thought of as an effect of other aspects of the biological system which are unknown or not possible to include in the model and may thus affect the system in a partly unpredictable way.

If a deterministic model based on ODEs is used to model such a system, these different sources of random or uncontrollable error can lead to significant systematic deviations in the model predictions which will result in correlated residuals with an ODE based model. This could be modelled using some form of auto-regressive process for the residuals, but only modelling this with SDEs will allow to identify in which parts of the model the randomness is occuring. The differences between ODE and SDE based models are further discussed in paper A.

Dynamic models based on SDEs can be incorporated in a mixed effects framework to handle larger overall differences between e.g. individuals, study centers, batches, and other structures in the data. The estimation of mixed models with SDEs is handled using the likelihood function in (4.20) in combination with the Laplace approximation.

As discussed there are strong arguments for modelling using SDEs in biological systems, but until recently there has not been software available designed for mixed models based on SDEs. It has been shown in Tornøe et al. (2005) that it is possible to perform the estimation in NONMEM with a model specific implementation of the extended Kalman filter by modifying the NONMEM data file and control stream. This approach is mainly aimed at experimentation with simple models as it somewhat complex to setup. This has led to the development

of the package PSM (Population Stochastic Modelling) for R which is able to estimate models in this general framework. The package is described in paper C. The package is developed with an extension of NONMEM in mind, and is thus like NONMEM designed only for data with a single level of grouping using the likelihood function in (2.6). The first stage model is thus given as $p_1(\mathcal{Y}_{in_i}|\boldsymbol{b}_i, \boldsymbol{\theta})$ where $\mathcal{Y}_{in_i}$ denotes the n_i observations and $\boldsymbol{b}_i$ the random effects for individual i. As mentioned, for an SDE based model this first stage likelihood function is evaluated using (4.20) where $\boldsymbol{b}_i$ and $\boldsymbol{\theta}$ together make up the parameters defining the individual SDE model.

Given that PSM is designed for one level of grouping it is not possible to handle crossed random effects that may arise from individuals being treated at different study centers or receiving treatment with drugs from different batches. Although the dynamical model based on SDEs will typically be set up for modelling single individuals this may still be combined with a model with the crossed random effects by letting each individual joint likelihood depend on the full vector of random effects. In this case the first stage model is defined as $p_1(\mathcal{Y}|\boldsymbol{b}, \boldsymbol{\theta}) = \prod_i p(\mathcal{Y}_{in_i}|\boldsymbol{b}, \boldsymbol{\theta})$ where $\mathcal{Y}$ denotes all observations and $p(\mathcal{Y}_{in_i}|\boldsymbol{b}, \boldsymbol{\theta})$ is again found using (4.20). The marginal likelihood requires an integration over the complete vector of random effects $\boldsymbol{b}$ as given in (2.4) which is evaluated using the multivariate Laplace approximation in (2.12). This application again emphasizes the great flexibility using the full multivariate Laplace approximation for mixed modelling.

4.6 Applications of SDE based models

The SDE based mixed modelling approach has been applied to a problem of estimating the insulin secretion rate and the extraction rate of insulin in the liver for patients with type II diabetes. This form of diabetes is caused by a reduced production of insulin together with a decreased sensitivity of the cell in the body to use insulin and both the insulin secretion rate and liver extraction rate are thus important to measure.

The data used for the analysis has been obtained from 12 type II diabetic patients (Degn et al., 2004). The patients were served three standardized meals over a 24 hour period and 35 samples of both insulin and C-peptide concentrations were taken during this period. C-peptide is a peptide that is made when pro-insulin is split into C-peptide and insulin and is thus produced in equimolar amounts. C-peptide is not extracted by the liver as insulin is and thus measuring the concentration of C-peptide in the blood is a more direct indication of insulin secretion since liver extraction does not have to be taken into account. The concentration of C-peptide in the blood is known to be well described by a two compartment model with rate constants given in Cauter et al. (1992). With this model given together with measurement of the C-peptide concentration it

is thereby possible to estimate the insulin secretion rate.

In paper A a method is suggested where the insulin secretion rate is modelled as a continuous random walk process represented as Brownian motion. This gives a relatively model free description of the secretion rate which can be estimated based on the data using the filtering approach for SDE models described above. The model includes mixed effects to handle individual differences and is estimated using an early prototype of PSM. Also a model for estimation of the liver extraction rate is presented using the combined measurements of insulin and C-peptide. The liver extraction is often modelled as a constant but with the SDE based approach it is possible to track time changes of this parameter in the model.

In paper C the model for the insulin secretion is extended to depend on meal times and estimated in the final version of PSM. A main benefit from estimating the secretion rate in the SDE based framework is that it seperates the measurement error from the estimate of the secretion rate and also that the probabilistic description makes it possible to provide confidence limits to estimates of the secretion.

Although not directly stated, SDE models also appear indirectly in other more common applications. The continuous auto-regressive (CAR) model in (2.13) used for the residuals of the orange tree data discussed in Section 2.4 implicitly states that the model residuals are generated by an Ornstein-Uhlenbeck (OU) process with mean 0. The OU-process is the solution to the stochastic differential equation

$$d\epsilon_t = -\phi\epsilon_t\, dt + \sigma\, d\omega_t \tag{4.21}$$

which has Gaussian conditional densities

$$\mathsf{E}[\epsilon_t|\epsilon_s] = \epsilon_s \exp(-\phi(t-s)) \tag{4.22}$$

$$\mathsf{V}[\epsilon_t|\epsilon_s] = \frac{\sigma^2(1-\exp(-2\phi(t-s)))}{2\phi} \tag{4.23}$$

for $s < t$ which can be derived using the Fokker-Planck equation (Iacus, 2008). The unconditional distribution is seen to be $\epsilon_t \sim N(0, \sigma^2/(2\phi))$ by letting $t \to \infty$ and the auto-covariance is given as $\mathrm{cov}(\epsilon_s, \epsilon_t) = \sigma^2/(2\phi)\exp(-\phi|t-s|)$ as also shown in (2.13) in a slightly different parametrization. The OU-process is called a mean reverting process due to (4.22) or sometimes colored noise due to the exponentially decreasing auto-correlation as opposed to a zero auto-correlation for the idealized white noise process.

This relation of the CAR residual correlation structure with the OU-process shows how it should be interpreted. The residuals for each tree can be seen as sampled directly from a realization of an OU-process where the process describes the difference between model prediction and the actual tree height. There is thus no independent measurement error in the model. This could be included by introducing a measurement equation as $e_k = \epsilon_{t_k} + z_k$ where $z_k \sim N(0, \sigma_z^2)$.

The effect is that measurements taken at (almost) coinciding time points will not have a correlation of 1 but instead only $\sigma^2/(\sigma^2 + \sigma_z^2)$. This is also called a nugget-effect (Matheron, 1962) and can easily be included directly in the residual correlation structure in the NLMM by adding σ_z^2 to the diagonal in $\mathbf{\Sigma}$ in (2.5).

Chapter 5

Conclusion

A central part of the statistical methods used in this thesis are based on the Markov property. Pharmacodynamic models can be described as hidden Markov processes in continuous state space by using stochastic differential equations to model random biological variations and fluctuations from unknown factors in the system. Sleep processes based on a classification of sleep states can similarly be modelled as a Markov process which instead evolves in a discrete state space. Although the response types in these two applications are fundamentally different the statistical models based on the Markov property show up as a strong method in both cases.

The work with pharmacodynamic models including SDEs in this project illustrates some of the benefits using this approach in comparisons to using models solely based on ODEs. In particular an application is presented which uses the SDE approach to estimate both insulin secretion and the liver extraction rates of insulin for patients with type II diabetes. This is a powerful method made available by SDEs that is generally applicable to estimation of unknown inputs or tracking of time varying parameters in both linear and nonlinear models. Today investigation of models based on SDEs within this area is mainly done for research purposes, and this may to a large part be due to the lack of software supporting estimation of SDE based models. This has been addressed during this project with the development of a software package (PSM) for this purpose, and the hope is that this will open up for an increased application of the SDE based approach in the future within pharmacodynamic modelling.

Within sleep modelling a Markov based model has been used to model the

sleep hypnogram, which is the time series observations of sleep states. The present work has focused on a continuous time representation which is argued to give the most appropriate representation of the process, as sleep is known to evolve in continuous time. The main contribution within this area has been the development of a non-parametric estimation method which is able to give both a more robust and accurate estimate of the time variations of the parameters in the model compared to other more simple approaches. Being based on non-parametric methods makes it entirely data driven and it can thus be used to give valuable detailed insight into the time dependence of the sleep process. This can be used to establish a direct relation with sleep effect to the concentration profile of a drug, and thereby give an early indication of how the drug interacts with the body. This approach has been illustrated based on a sleep study in rats where both the PK profile and sleep effects have been considered in combination.

Both the Markov models for sleep and for pharmacodynamic models based on SDEs can be used in a mixed modelling approach to handle variation between groups in the data. If data has a crossed grouping structure (such as for individuals belonging to different study centers) this poses a difficult estimation problem which cannot easily be handled by standard software. It is proposed and demonstrated how such models instead can be estimated by a direct implementation of the marginal likelihood function, which avoids these restrictions on the grouping structure for mixed modelling.

Bibliography

ADMB Project (2009). AD Model Builder Project. *http://admb-project.org/*.

Anderson, N. J., S. L. Garson, S. V. Fox, S. M. Doran, B. Ebert, and J. J. Renger (2006). SC & PO Gaboxadol cause dose-related increase in delta sleep without EEG hyper-synchrony. *Society for Neuroscience*. November 2006 meeting. 157.30/V16.

Bates, D. M. and D. G. Watts (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley, New York.

Beal, S. and L. Sheiner (1980). The NONMEM System. *The American Statistician 34*, 118–119.

Beal, S. L. and L. B. Sheiner (2004). *NONMEM® Users Guide*. NONMEM Project Group, University of California, San Francisco.

Bladt, M. and M. Sørensen (2005). Statistical inference for discretely observed markov jump processes. *Journal of the Royal Statistical Society Series B 67*(3), 395–410.

Blume, J. and J. F. Peipert (2003). What your statistician never told you about p-values. *The Journal of the American Association of Gynecologic Laparoscopists 10*, 439–444.

Brodal, P. (2001). *Sentralnervesystemet* (3rd ed.). Oslo, Norway: Universitetsforlaget.

Cauter, E. V., F. Mestrez, J. Sturis, and K. S. Polonsky (1992). Estimation of insulin secretion rates from c-peptide levels. comparison of individual and standard kinetic parameters for c-peptide clearance. *Diabetes 41*, 368–77.

Cox, D. R. and H. D. Miller (1965). *The Theory of Stochastic Processes*. Methuen & Co Ltd.

Cuthbert, J. R. (1973). The logarithm function for finite-state markov semigroups. *Journal of the London Mathematical Society 6*, 524–532.

Degn, K. B., C. B. Juhl, J. Sturis, G. Jakobsen, B. Brock, V. Chandramouli, J. Rungby, B. R. Landau, and O. Schmitz (2004). One Week's Treatment With the Long-Acting Glucagon-Like Peptide 1 Derivative Liraglutide (NN2211) Markedly Improves 24-h Glycemia and alpha- and beta-Cell Function and Reduces Endogenous Glucose Release in Patients with Type 2 Diabetes. *Diabetes 53*, 1187–1194.

Diggle, P. J. (1988). An approach to the analysis of repeated measurements. *Biometrics 44*, 959–71.

Draper, N. R. and H. Smith (1981). *Applied Regression Analysis* (2nd ed.). Wiley, New York.

du Toit, S. H. C. and R. Cudeck (2009). Estimation of the nonlinear random coefficient model when some random effects are separable. *Psychometrika 74*, 65–82.

Eaton, J. W. (2002). *GNU Octave Manual*. Network Theory Limited.

Elfving, G. (1937). Zur theorie der markoffschen ketten. *Acta Societatis Scientiarum Fennicae A 2*, 1–17.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and Its Applications*. Number 66 in Monographs on Statistics and Applied Probability. Chapman & Hall.

Forehand, C. J. (2003). *Medical Physiology* (2nd ed.)., Chapter 7, Integrative Functions of the Nervous System. Lippincott Williams & Wilkins.

Gabrielsson, J. and D. Weiner (2006). *Pharmacokinetic and Pharmacodynamic Data Analysis: Concepts and Applications* (4th ed.). Kristianstads Boktryckeri.

Gard, T. (1988). *Introduction to Stochastic Differential Equations*. New York: Marcel Dekker.

Goodman, S. N. (1999). Toward evidence-based medical statistics. 1: The p value fallacy. *Annals of Internal Medicine 130*, 995–1004.

Griewank, A. (2000). *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Frontiers in applied mathematics. Philadelphia, PA: SIAM.

Grube, G., A. Flexer, and G. Dorffner (2002). Unsupervised continuous sleep analysis. *Methods & Findings in Experimental & Clinical Pharmacology 24*, 51–56.

Himanen, S.-L. and J. Hasan (2000). Limitations of Rechtschaffen and Kales. *Sleep Medicine Reviews 4*, 149–167.

Himanen, S.-L., A. Saastamoinen, and J. Hasan (1999). Increasing the temporal resolution and stage specificity by visual adaptive scoring (vas) - a preliminary description. *Sleep and Hypnosis 1*, 22–28.

Iacus, S. M. (2008). *Simulation and Inference for Stochastic Differential Equations - With R Examples*. Springer Series in Statistics. New York, USA: Springer.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. New York, USA: Academic Press, Inc.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering 82*, 35–45.

Karlsson, M. O., R. C. Schoemaker, B. Kemp, A. F. Cohen, J. M. van Gerven, B. Tuk, C. C. Peck, and M. Danhof (2000). A pharmacodynamic markov mixed-effects model for the effect of temazepam on sleep. *Clin Pharmacol Ther 68*, 175–188.

Kemp, B. and H. A. C. Kamphuisen (1986). Simulation of human hypnograms using a markov chain model. *Sleep 9*, 405–414.

Kingman, J. F. C. (1962). The imbedding problem for finite markov chains. *Z. Wahrscheinlichkeitstheorie 1*, 14–24.

Kristensen, N. R. and H. Madsen (2003). Continuous time stochastic modelling - ctsm 2.3 mathematics guide. Technical report, Technical University of Denmark.

Lee, H. B. and M. D. Blaufox (1985). Blood volume in the rat. *Journal of Nuclear Medicine 25*, 72–76.

Lindstrom, M. J. and D. M. Bates (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics 46*, 673–687.

Madsen, H. (2008). *Time Series Analysis* (1st ed.). Texts in Statistical Science. Chapman & Hall/CRC.

Madsen, H., H. Spliid, and P. Thyregod (1985). Markov models in discrete and continuous time for hourly observations of cloud cover. *Journal of Applied Meteorology 24*(7), 629–639.

Matheron, G. (1962). *Traite de Geostatistique Appliquee*, Volume I of *Memoires du Bureau de Recherches Geologiques et Minieres*. Paris: Editions Technip, Paris.

McCulloch, C. E. and S. R. Searle (2001). *Generalized, Linear, and Mixed Models*. John Wiley & Sons, Inc.

Millar, R. B. (2004). Simulated maximum likelihood applied to non-gaussian and nonlinear mixed effects and state-space models. *Australian & New Zealand Journal of Statistics 46*, 543–554.

Moler, C. B. and C. F. Van Loan (1978). Nineteen dubious ways to compute the exponential of a matrix. *SIAM Review 20*, 801–836.

Nielsen, H. B. (2000). UCMINF - an algorithm for unconstrained, nonlinear optimization. Technical Report IMM-REP-2000-19, Informatics and Mathematical Modelling (IMM), Technical University of Denmark.

Nielsen, H. B. and S. B. Mortensen (2008). *ucminf: General-purpose unconstrained non-linear optimization*. R package version 1.0-6.

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization* (2nd ed.). Springer Series in Operations Research. Springer.

Øksendal, B. (2007). *Stochastic Differential Equations - An Introduction with Applications* (6th ed.). Springer.

Pawitan, Y. (2001). *In All Likelihood: modelling and inference using the likelihood*. Oxford University Press.

Pinheiro, J. C. and D. M. Bates (1995). Approximations to the nonlinear mixed-effects model. *Jounal of Computational and Graphical Statistics 4*, 12–35.

Pinheiro, J. C., D. M. Bates, S. DebRoy, D. Sarkar, and the R Core team (2008). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-90.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rechtschaffen, A. and A. Kales (1968). *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. Washington DC.

Rowland, M. and T. N. Tozer (1997). *Clinical Pharmacokinetics - Concepts and Applications* (3rd ed.). Lippincott Williams & Wilkins.

Royall, R. (1997). *Statistical evidence - A likelihood paradigm*. Monographs on statistics and applied probability 71. Chapman & Hall/CRC.

SAS Institute Inc. (2004). *SAS/Stat 9.1 User's Guide*. Cary, NC: SAS Institute Inc.

Sidje, R. B. (1998). EXPOKIT. A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software 24*(1), 130–156.

Skaug, H. J. and D. A. Fournier (2006). Automatic approximation of the marginal likelihood in non-gaussian hierarchical models. *Computational Statistics & Data Analysis 51*, 699–709.

Tornøe, C. W., R. V. Overgaard, H. Agersø, H. A. Nielsen, H. Madsen, and E. N. Johnson (2005). Stochastic differential equations in NONMEM®: Implementation, application, and comparison with ordinary differential equations. *Pharmaceutical Research 22*, 1247–1258.

Turelli, M. (1977). Random environments and stochastic calculus. *Theoretical Population Biology 12*, 140–178.

Vonesh, E. F. (1996). A note on the use of laplace's approximation for nonlinear mixed-effects models. *Biometrika 83*, 447–52.

Wafford, K. A. and B. Ebert (2006). Gaboxadol – a new awakening in sleep. *Current Opinion in Pharmacology 6*, 30–36.

Walsh, J. K., S. Deacon, D.-J. Dijk, and J. Lundahl (2007). The selective extrasynaptic gabaa agonist, gaboxadol, improves traditional hypnotic efficacy measures and enhances slow wave activity in a model of transient insomnia. *Sleep 30*, 593–602.

Wang, Y. (2007). Derivation of various NONMEM estimation methods. *Journal of Pharmacokinetics and Pharmacodynamics 34*, 575–593.

Ward, R. C. (1977). Numerical computation of the matrix exponential with accuracy estimate. *SIAM Journal on Numerical Analysis 14*, 600–610.

Wolfinger, R. D. and X. Lin (1997). Two taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis 25*, 465–490.

Zung, W. W. K., T. H. Naylor, D. Gianturco, and W. P. Wilson (1965). A Markov chain model of sleep EEG patterns. *Electroencephalography and Clinical Neurophysiology 19*, 105.

Appendix A

Paper A

Title:
A Matlab Framework for Estimation of NLME Models using Stochastic Differential Equations: Applications for estimation of insulin secretion rates.

Authors:
S. B. Mortensen, S. Klim, B. Dammann, N. R. Kristensen, H. Madsen, and R. V. Overgaard.

Published in:
Journal of Pharmacokinetics and Pharmacodynamics 34, pp. 623-42 (2007).

J Pharmacokinet Pharmacodyn (2007) 34:623–642
DOI 10.1007/s10928-007-9062-4

A matlab framework for estimation of NLME models using stochastic differential equations

Applications for estimation of insulin secretion rates

Stig B. Mortensen · Søren Klim · Bernd Dammann · Niels R. Kristensen · Henrik Madsen · Rune V. Overgaard

Received: 17 November 2006 / Accepted: 27 April 2007 / Published online: 15 June 2007

Abstract The non-linear mixed-effects model based on stochastic differential equations (SDEs) provides an attractive residual error model, that is able to handle serially correlated residuals typically arising from structural mis-specification of the true underlying model. The use of SDEs also opens up for new tools for model development and easily allows for tracking of unknown inputs and parameters over time. An algorithm for maximum likelihood estimation of the model has earlier been proposed, and the present paper presents the first general implementation of this algorithm. The implementation is done in Matlab and also demonstrates the use of parallel computing for improved estimation times. The use of the implementation is illustrated by two examples of application which focus on the ability of the model to estimate unknown inputs facilitated by the extension to SDEs. The first application is a deconvolution-type estimation of the insulin secretion rate based on a linear two-compartment model for C-peptide measurements. In the second application the model is extended to also give an estimate of the time varying liver extraction based on both C-peptide and insulin measurements.

Keywords Non-linear mixed-effects modelling · SDE · Kalman smoothing · Deconvolution · State-estimation · Parameter tracking · MatlabMPI · PK/PD

S. B. Mortensen (✉)· S. Klim · B. Dammann · H. Madsen
Informatics and Mathematical Modelling, Technical University of Denmark,
Lyngby, Denmark
e-mail: sbm@imm.dtu.dk

S. Klim · N. R. Kristensen · R. V. Overgaard
Novo Nordisk A/S, Bagsvaerd, Denmark

Introduction

The non-linear mixed-effects (NLME) model based on ordinary differential equations (ODEs) is a widely used method for modelling pharmacokinetic/pharmacodynamic (PK/PD) data [1], since the model enables the variation to be split into inter- and intra-individual variation. It is, however, a well known problem that this model class has a too restricted residual error structure, as it assumes that the residuals are uncorrelated white noise. This assumption applies well to the expected distribution of assay error, but it is unfortunately a crude simplification to assume that the assumption also applies to the remaining sources of error [2]. Other important sources of error may arise from structural model mis-specification or unpredictable random behavior of the underlying process, which both result in serially correlated residual errors. Previous work with simulation of more complex error structures has shown that ignoring the serial correlation may lead to biased estimates of the variance components of the model or all population parameter estimates depending on the error structure [3].

A powerful way to deal with these problems is to introduce stochastic differential equations (SDEs) in the model setup. SDEs are an extension to ODEs and facilitate the ability to split the intra-individual error into two fundamentally different types: (1) *serially uncorrelated measurement error*, which is typically mainly caused by assay error and (2) *system error*, which may be caused by model mis-specifications, simplifications or true random behavior of the system.

Apart from providing a statistically more adequate model setup, the SDEs also allow new tools for the modeller. The SDE approach results in a quantitative estimate of the amount of system and measurement noise, and it can therefore also be used as a tool for model validation. If no significant system noise is found to be present, this indicates that the proposed model structure gives a suitable description of the data. However, if significant system noise is found, it can be estimated and may be used to identify a possible remaining model structure, since aspects which are not explicitly modelled will give rise to system noise. It is important to emphasize that this relation does not hold the other way around, since system noise may also arise from true unmodellable random behavior of the system, and estimated system noise may thus not be seen as evidence of an insufficient model structure. A detailed iterative scheme for model development based on SDEs has been described in [4]. Another important advantage of the SDE approach is the inherent confidence intervals for system states. This is facilitated by the estimation of system noise, and thus follows as a natural part of the model specification.

Several programs exist for modelling based on SDEs. The first implementations focused on single subject modelling, such as Continuous Time Stochastic Modelling (CTSM) [5]. CTSM is in fact also able to use multiple individuals for estimation of structural parameters, but this is done using a naive pooled likelihood function where no inter-individual variance components are estimated. Later research has also made it possible to include SDEs in population modelling by using an approximation algorithm of the likelihood function with SDEs for the widely used NLME model. This algorithm is described in [6] and is based on the use of the Extended Kalman Filter to estimate conditional densities of each observation to form the individual likelihood function. The population likelihood function is then approximated based on the

first-order conditional estimation (FOCE) method. It has been shown in [7] how this algorithm for estimation of SDEs can be used in NONMEM [8], but this is by no means a trivial programming task to set up for a given model. It requires a modified data file and an implementation of the Kalman filter within the NONMEM control stream. Moreover, the NONMEM implementation cannot be used to form Kalman smoothing estimates, which is an important feature of the SDE approach, where all data is used to give optimal estimates at each sampling point.

This paper will present the first prototype implementation of a general software tool for estimation of NLME models based on SDEs. The implementation has been made in Matlab and it makes experimentation with the new modelling approach readily available. The flexibility of the modelling approach will be demonstrated by two examples of applications. In the first example the model is used for stochastic deconvolution to estimate insulin secretion rates in 12 type II diabetic patients and in the second example the model is used to estimate/track the time variant behavior of the liver extraction rate for the same individuals.

Theory

This section contains an overview of the theory for population modelling using NLME models based on SDEs. It will present the state space model for individual modelling and how this can be extended to incorporate SDEs. The parameters of the population model are estimated with a maximum likelihood (ML) approach by first defining an individual likelihood function, which forms the basis for the population likelihood function. The individual likelihood function is evaluated on the basis of the Extended Kalman Filter (EKF), and this will also be outlined. A more detailed description of the estimation algorithm can be found in [6]. To ease notation, all vectors and matrices are written using a bold font.

A mixed-effects model is used to describe data with the following general structure

$$\boldsymbol{y}_{ij}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, n_i \tag{1}$$

where $\boldsymbol{y}_{ij}$ is a vector of measurements at time t_{ij} for individual i, N is the number of individuals and n_i is the number of measurements for individual i. Note that the number of measurements for each individual may vary. In a mixed-effects model the variation is split into intra-individual variation and inter-individual variation, which is modelled by a first and second stage model.

First stage model

The first stage model for an NLME model with ODEs can be written in the form of a state space model. A state space model consists of two parts, namely a set of continuous state equations defining the dynamics of the system and a set of discrete measurement equations, which defines a functional relationship between the states of the system and the measurements obtained. In the general form the state space equations are written as

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\phi}_i)dt \tag{2}$$

$$\boldsymbol{y}_{ij} = \boldsymbol{h}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, t_{ij}, \boldsymbol{\phi}_i) + \boldsymbol{e}_{ij} \tag{3}$$

where t is the continuous time variable and the states of the model and the optional inputs at time t are denoted $\boldsymbol{x}_t$ and $\boldsymbol{u}_t$ respectively. The input $\boldsymbol{u}_{ij}$ is typically frequently sampled covariates such as body temperature etc. which may affect the system, or a variable indicating an interaction with the system such as an intravenous infusion. Both the state, measurement and input can be multi-dimensional, and are in such cases thus represented by a vector at time t_{ij}. The individual model parameters are denoted $\boldsymbol{\phi}_i$ and finally $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ are the two possibly non-linear functions defining the model. Measurements are assumed observed with an uncorrelated Gaussian measurement error. The variance of the error may depend on both state, input, time and individual parameters, that is $\boldsymbol{e}_{ij} \in N(\boldsymbol{0}, \boldsymbol{\Sigma}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, t_{ij}, \boldsymbol{\phi}_i))$.

It is important to draw attention to the concept of states, as this is essential to the understanding of the model setup. States are generally not directly observable or at best only observable through measurement noise. The actual relation between measurements and states is defined in the measurement equation by the function $\boldsymbol{h}(\cdot)$. A state can represent many different aspects of the system of interest, e.g. concentrations or amounts in compartments, a volume, a parameter with unknown time varying behavior, or an input to the system that we wish to estimate. The state space formulation is thus a very flexible form of model specification, and the use of the state space model will be illustrated with the applications presented later on in this paper.

Extending the first stage model with SDEs

In the ordinary state space model, noise is only allowed to enter through the measurement equation, see Eq. 3. The result is that error due to model mis-specification or true random fluctuations of the states is absorbed into the measurement error term and hence may give rise to correlated residuals. To allow for error to originate from the system specification, a stochastic term is added to the system equation. This results in a stochastic state space equation defined as follows

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\phi}_i)dt + \boldsymbol{\sigma}_\omega(\boldsymbol{u}_t, t, \boldsymbol{\phi}_i)d\boldsymbol{\omega}_t \tag{4}$$

$$\boldsymbol{y}_{ij} = \boldsymbol{h}(\boldsymbol{x}_{ij}, \boldsymbol{u}_{ij}, t_{ij}, \boldsymbol{\phi}_i) + \boldsymbol{e}_{ij} \tag{5}$$

where $\boldsymbol{\omega}_t$ is a standard Wiener process defined by $\boldsymbol{\omega}_{t_2} - \boldsymbol{\omega}_{t_1} \in N(\boldsymbol{0}, |t_2 - t_1|\boldsymbol{I})$. The entire part $\boldsymbol{\sigma}_\omega(\boldsymbol{u}_t, t, \boldsymbol{\phi}_i)d\boldsymbol{\omega}_t$ is called the diffusion term and describes the stochastic part of the system and $\boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\phi}_i)dt$ is called the drift term and describes the deterministic part. Together the drift and diffusion terms define the stochastic dynamics of the system.

By looking at the formulation of the extended first stage model, it is seen that noise is now allowed to enter in two places, namely as system noise via the diffusion term and as measurement noise. It is noted that if no system noise is present, the model will reduce into the standard ODE case, and this also ensures that physiological interpretation of structural parameters is preserved with the use of an SDE model.

Second stage model

The second stage model for the individual parameters describes the variation of the individual parameters $\boldsymbol{\phi}_i$ between individuals and can be defined in a number of ways, each with different properties. In the present work it has been chosen to use

$$\boldsymbol{\phi}_i = g(\boldsymbol{\theta}, \boldsymbol{Z}_i) \cdot \exp(\boldsymbol{\eta}_i) \tag{6}$$

where $\boldsymbol{\eta}_i$ is the multivariate random effect parameter for the ith individual, which is assumed Gaussian distributed with mean zero and covariance $\boldsymbol{\Omega} : \boldsymbol{\eta}_i \in N(\mathbf{0}, \boldsymbol{\Omega})$. The fixed effect parameter of the NLME model is $\boldsymbol{\theta}$, which is also sometimes referred to as the structural parameter or population parameter. The second stage model in Eq. 6 includes an optional covariate $\boldsymbol{Z}_i$. This can be used to include individually measurable covariates such as height, weight etc. that could affect $\boldsymbol{\phi}_i$. The chosen formulation of the 2nd stage model restricts variations in $\boldsymbol{\eta}_i$ from changing the sign of $g(\boldsymbol{\theta}, \boldsymbol{Z}_i)$ which is typically an advantage as $\boldsymbol{\phi}_i$ may be used as parameter for a variance or other sign-sensitive parameters. Moreover the resulting distribution of the individual parameters is log-Gaussian, as is often the case when dealing with PK/PD models. The second stage model in Eq. 6 may easily be replaced if other model structures are needed, and this can be done without yielding any changes to the final population likelihood function as long as $\boldsymbol{\eta}_i$ is still assumed to have a Gaussian distribution.

Maximum likelihood estimation of the NLME model with SDEs

The full set of parameters to be estimated for the final NLME model with SDEs are the matrices $\boldsymbol{\Sigma}$, $\boldsymbol{\sigma}_\omega$, $\boldsymbol{\Omega}$ and the fixed effect parameters in the vector $\boldsymbol{\theta}$. The three matrices are usually fixed to some degree so that only the diagonals or other partial structure remains to be estimated.

The estimation of model parameters is based on a first stage likelihood function, which is formed as a product of probabilities for each measurement. Due to the assumption of correlated residuals with the inclusion of the Wiener process, it is necessary to condition on the previous measurements to define the probability density of each measurement. In the approach chosen here, this is done by assuming that the conditional densities for the states are Gaussian and thus fully described by the state-prediction and the state prediction variance for each observation. These can be found using the Extended Kalman filter, which gives the unbiased minimum variance estimate of the evolution of the model states [9]. This will hold exactly for the linear case but only as an approximation in the non-linear case. The assumptions for the EKF can be examined by testing for a Gaussian distribution of the residuals and by testing for correctness of the estimated SDEs [10]. The prediction from the EKF is defined by

$$\hat{\boldsymbol{y}}_{i(j|j-1)} = E(\boldsymbol{y}_{ij}|\boldsymbol{\phi}_i, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{u}_i, \mathcal{Y}_{i(j-1)}) \tag{7}$$

$$\boldsymbol{R}_{i(j|j-1)} = V(\boldsymbol{y}_{ij}|\boldsymbol{\phi}_i, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{u}_i, \mathcal{Y}_{i(j-1)}) \tag{8}$$

where $\mathcal{Y}_{ij} = [\boldsymbol{y}_{i1}, \ldots, \boldsymbol{y}_{ij}]$ and this gives the conditional distribution of the one-step prediction error

$$\boldsymbol{\epsilon}_{ij} = \boldsymbol{y}_{ij} - \hat{\boldsymbol{y}}_{i(j|j-1)} \in N(\boldsymbol{0}, \boldsymbol{R}_{i(j|j-1)}) \tag{9}$$

The first stage likelihood function is calculated as the simultaneous density function for the ith individual

$$p_1(\mathcal{Y}_{in_i}|\boldsymbol{\phi}_i, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{u}_i) = \left(\prod_{j=2}^{n_i} p(y_{ij}|\mathcal{Y}_{i(j-1)}, \cdot)\right) p(y_{i1}|\cdot) \tag{10}$$

$$\approx \prod_{j=1}^{n_i} \frac{\exp\left(-\frac{1}{2}\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)}^{-1}\boldsymbol{\epsilon}_{ij}\right)}{\sqrt{|2\pi\,\boldsymbol{R}_{i(j|j-1)}|}} \tag{11}$$

where conditioning on $\boldsymbol{\phi}_i$, $\boldsymbol{\Sigma}$, $\boldsymbol{\sigma}_\omega$ and $\boldsymbol{u}_i$ is denoted "$\cdot$".

Based on the first and second stage model density functions, the full NLME likelihood function can now be defined. The second stage distribution is simply a multivariate Gaussian density denoted $p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega})$, and combining this with the first stage distribution results in the population likelihood function

$$L(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{\Omega}) = \prod_{i=1}^{N} \int p_1(\mathcal{Y}_{in_i}|\boldsymbol{\eta}_i, \boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{u}_i) p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) d\boldsymbol{\eta}_i \tag{12}$$

$$= \prod_{i=1}^{N} \int \exp(l_i) d\boldsymbol{\eta}_i \tag{13}$$

where l_i is the a posteriori log-likelihood function for the random effects of the ith individual given by

$$l_i = -\frac{1}{2}\sum_{j=1}^{n_i}\left(\boldsymbol{\epsilon}_{ij}^T \boldsymbol{R}_{i(j|j-1)}^{-1}\boldsymbol{\epsilon}_{ij} + \log|2\pi\,\boldsymbol{R}_{i(j|j-1)}|\right) - \frac{1}{2}\boldsymbol{\eta}_i^T\boldsymbol{\Omega}^{-1}\boldsymbol{\eta}_i - \frac{1}{2}\log|2\pi\,\boldsymbol{\Omega}| \tag{14}$$

The population likelihood function in Eq. 13 cannot be evaluated analytically, and therefore l_i is approximated by a second-order Taylor expansion, where the expansion is made around the value $\hat{\boldsymbol{\eta}}_i$ that maximizes l_i. At this optimum the first derivative $\boldsymbol{\nabla} l_i\big|_{\hat{\boldsymbol{\eta}}_i} = 0$ and the population likelihood function therefore reduces to

$$L(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_\omega, \boldsymbol{\Omega}) \approx \prod_{i=1}^{N} \left|\frac{-\boldsymbol{\Delta} l_i}{2\pi}\right|^{-\frac{1}{2}} \exp(l_i)\Big|_{\hat{\boldsymbol{\eta}}_i} \tag{15}$$

as shown in Appendix. The approximation of the 2nd derivative $\boldsymbol{\Delta} l_i$ is done using the FOCE method, as it is also normally done in the NLME model based on ODEs. The

objective function for parameter estimation is chosen as the negative log-likelihood function given as

$$-\log L(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_{\omega}, \boldsymbol{\Omega}) \approx \sum_{i=1}^{N} \left(\frac{1}{2} \log \left| \frac{-\boldsymbol{\Delta} l_i}{2\pi} \right| - l_i \right) \tag{16}$$

Kalman filtering

The Extended Kalman Filter plays a central role for working with the NLME model with SDEs as seen from the previous section. Therefore a brief introduction to the EKF will be given here, as well as to the three new types of state estimates made available. For a detailed description of the EKF algorithm please refer to [5,6,11,12].

For linear state-estimation problems the Kalman Filter will give an unbiased minimum variance state estimate. The solution can be derived explicitly using simple linear algebra, and hence the algorithm runs efficiently in a computer implementation. For non-linear problems it is necessary to use another method for state-estimation like that obtained by the Extended Kalman Filter, which has been used here. The Extended Kalman Filter is for the main part identical to the Kalman Filter, except for the state prediction which requires a solution to the non-linear differential system equations. This solution is obtained by a point-wise first-order approximation and therefore, for non-linear systems, the EKF will only provide an approximate minimum variance estimate of the states. The EKF also runs slower due to the need for a numerical algorithm to solve the non-linear differential equations.

The Kalman Filter is a two-part algorithm consisting of *prediction* and *updating*, which iterates through all observations. In the prediction part the current estimated states and covariances are used to create predictions of the two first moments of the state and observation to a time point t_{ij} given the information at time $t_{i(j-1)}$. These predictions are denoted $\hat{\boldsymbol{x}}_{i(j|j-1)}$, $\hat{\boldsymbol{P}}_{i(j|j-1)}$, $\hat{\boldsymbol{y}}_{i(j|j-1)}$ and $\hat{\boldsymbol{R}}_{i(j|j-1)}$, respectively. Updating is performed at measurement time points, where the states and covariances are updated accordingly.

The updating is based on a compromise between the observation and current model state. In a situation where the model is good but the observations are dominated by measurement error, the state estimate should rely more on the model as opposed to fitting the observations. On the other hand, if the model is incomplete the states should rely more on the observations than the model. This trust in model versus observations is balanced by the Kalman gain, which is dependent on the magnitude of system noise $\boldsymbol{\sigma}_{\omega}$ and observation noise $\boldsymbol{\Sigma}$.

The initial conditions of the state and state covariance ($\hat{\boldsymbol{x}}_{i(1|0)}$ and $\hat{\boldsymbol{P}}_{i(1|0)}$) need to be specified for the Kalman filtering algorithm. The initial state can either be fixed or included in the likelihood function, whereas $\hat{\boldsymbol{P}}_{i(1|0)}$ for this implementation has been chosen to be estimated as the integral of the Wiener process and system dynamics over the first sample interval in accordance with the method used in [11].

A key feature of the SDE approach to population modelling is the ability to give improved estimates of the system states given the individual parameters and also to

provide confidence bands for the states. Confidence bands at a timepoint t are directly given by the estimated state covariance matrix $\hat{\boldsymbol{P}}_{i(t|\ldots)}$ from the EKF, where t can be both at or between measurements. There are four types of state and state covariance estimates available when using the EKF, each of which differs in the way data is used. The four types are:

- Simulation estimate: $\hat{\boldsymbol{x}}_{i(j|0)}$, $\hat{\boldsymbol{P}}_{i(j|0)}$
  Provides an estimate of the state evolution for a repeated experiment, without updating based on measurements. This is an ODE-like estimate, but it also yields a confidence band for the state evolution.

- Prediction estimate: $\hat{\boldsymbol{x}}_{i(j|j-1)}$, $\hat{\boldsymbol{P}}_{i(j|j-1)}$
  The prediction is used here to give the conditional density for the next observation at time t_{ij} given the observations up to $t_{i(j|j-1)}$.

- Filtering estimate: $\hat{\boldsymbol{x}}_{i(j|j)}$, $\hat{\boldsymbol{P}}_{i(j|j)}$
  Best estimate at time t_{ij} given the observations up to time t_{ij}.

- Smoothing estimate: $\hat{\boldsymbol{x}}_{i(j|N)}$, $\hat{\boldsymbol{P}}_{i(j|N)}$
  Optimal estimate at time t_{ij} utilizing all observations both prior to and after time t_{ij}.

For a conventional ODE model the state is found by the simulation estimate, which is entirely given by the (possibly ML-estimated) initial state of the system. The covariance matrix for the states is $\mathbf{0}$ since no system noise is estimated. In other words the ODE model assumes that a new experiment will yield an identical outcome of the underlying system apart from observed measurement noise. By moving to SDEs, system noise is separated from measurement noise, thereby enabling the model to provide confidence bands for the realization of the states in a new experiment. By improving the model, the confidence bands for the states will become narrower and theoretically be zero if the true model is used and no random fluctuations in system states are present.

With SDEs three new types of estimates, apart from the simulation estimate, also become available. In the present setup the prediction estimate is used to give conditional Gaussian densities to form the likelihood function. The filter estimate is the best obtainable state estimate during the experiment, where the subsequent observations are not present. The third type of state estimate is the smoothed estimate. This provides the optimal state and state covariance estimate ($\hat{\boldsymbol{x}}_{i(j|N)}$ and $\hat{\boldsymbol{P}}_{i(j|N)}$) based on all obtained observations, both prior and subsequent to the time of interest. The smoothed estimate is therefore often the natural estimate of choice when studying the behavior of the system in post hoc analysis.

Software implementation

The estimation algorithm outlined in the previous section has been implemented in a Matlab framework called population stochastic modelling (PSM). It is intended that this should work as a software prototype, in order to make further experimentation

with the model setup easily available. The program may be obtained by addressing an email to the corresponding author.

Features

The implementation is designed to handle any non-linear mixed effects models using SDEs based on the general model definition in the previous section. The model specification is achieved through a set of Matlab functions written in m-files. A complete model specification consists of state dynamics $\boldsymbol{f}$ and diffusion term magnitude $\boldsymbol{\sigma}_\omega$, output function $\boldsymbol{h}$ and uncertainty $\boldsymbol{\Sigma}$, derivatives of state $d\boldsymbol{f}/dt$ and output $d\boldsymbol{h}/dt$, initial state $\boldsymbol{x}_0$, second stage model $\boldsymbol{g}$ and finally a variance function $\boldsymbol{\Omega}$ for the random effects. Each function is prepared to use all input arguments as specified by the model definition.

The implementation has been made in two versions. The first is able to handle the general non-linear case, and is thus based on the use of an algorithm for solving the differential equations in the EKF. It has been chosen to use `ode15s`, which is a Matlab built-in ODE solver. The second version is only able to solve linear systems, which will run significantly faster since it is based on an explicit solution of the differential equations.

The population parameters are estimated by maximizing of the population likelihood function given in Eq. 15. Maximization is performed using a publicly available Matlab implementation `ucminf` of a gradient search BFGS method with soft-line search and trust-region type monitoring of step length [13]. For additional performance it is possible to guide the optimization by providing an initial guess and boundaries for the parameters. The implementation is also able to assess parameter variance and correlation based on a numerical approximation of the Hessian of the likelihood functions [14].

Implementation details

In the evaluation of the population likelihood function it is necessary to evaluate the individual a posteriori log-likelihood function for each individual at its optimum, since a Taylor expansion is made around this point. Hence for one evaluation of the population likelihood function an optimization must be performed on each individual likelihood function. These optimizations only share the given population parameters and are therefore evaluated independently. This observation can be used to employ the use of parallel computing, where the individual optimizations are distributed to a number of CPUs.

Matlab does not have the option for parallel computing by default[1], but this can be made possible using external software. MatlabMPI[2] is a package developed at MIT and it enables parallel computing in Matlab by creating a set of scripts that is executed

[1] A distributed toolbox for Matlab is under development by The MathWorks.

[2] J. Kepner, Parallel Programming with MatlabMPI, http://www.ll.mit.edu/MatlabMPI/, 2006.

Table 1 Computation times using parallel computing

CPUs	Time (s)	Reduced to (%)	CPU-time per individual (s)	Overhead per CPU (%)
1 serial	241.8	100.0	12.1	−−
1	242.4	100.2	12.1	0.2
2	128.3	53.0	12.8	5.8
3	101.7	42.0	15.3	20.7
4	72.0	29.7	14.4	16.0
5	66.0	27.3	16.5	26.7
10	50.0	20.6	25.0	51.6

in separate processes. MatlabMPI uses message passing but it was found faster to pass all data and parameters through files. The individual calculation extracts its unique part of data by using its identifier number. The individual log-likelihood result is passed back into the leader thread by proper message passing to avoid deadlocks or race conditions. A shared memory environment is beneficial as message passing is implemented through shared files.

In order to illustrate the effect of parallel computation for population modelling, a model was setup and estimated on the basis of simulated data for 20 subjects. The resulting computation time is found in Table 1 and it can be seen that the computation time is reduced to a little less than one-fifth of the original using five CPUs. For this example overhead begins to dominate when using more than five CPUs, however for more computationally intensive models, the benefit of adding more CPUs is expected to be less affected by overhead.

For non-linear models a significant part of the computation time is spent in the prediction part of the Extended Kalman Filter when solving the differential system equations. The prediction includes both state and state covariance, and these differential equations are coupled and must therefore be solved simultaneously. To account for this coupling, the two prediction equations have been collected into one system with a combined input vector Z which stores both the states and the covariance matrix. The symmetry in the covariance matrix is exploited so only the upper part is transferred, i.e.

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = \begin{pmatrix} \hat{\boldsymbol{x}}_{t|k} \\ U(\boldsymbol{P}_{t|k}) \end{pmatrix} \tag{17}$$

where $U()$ is a column vector containing the upper matrix. Conversion to the vector Z is then used in conjunction with `ode15s` and the output is converted back into a state vector and covariance matrix. The use of a single vector Z complies with the Matlab standard conventions for ODE solving algorithms, and the chosen algorithm `ode15s`may thus easily be substituted to suit the dynamic properties of a given model.

Validation of implementation

The implementation of PSM has been validated with CTSM and NONMEM. The comparison with CTSM has been used to verify correct implementation of the Kalman Filter and Extended Kalman Filter by comparison with CTSM's individual likelihood function. The comparison was based on a model using SDEs and showed identical outcomes from the two programs.

The comparison with NONMEM was done with a model based on ODEs in order to verify the population likelihood function. The comparison showed that PSM produces identical population parameter estimates and also identical estimates of the individual random effects parameters for four simulated data sets containing 2, 4, 10 and 20 subjects.

A final validation with NONMEM was done on the objective function value. The NONMEM objective function (l_{NM}) is advertised as $-2\log L$ but in fact it lacks a constant equal to the likelihood of the data. The PSM objective function (l_{PSM}) is $-\log L$ as seen in Eq. 16 and the relation thereby becomes $l_{NM} = 2{\cdot}l_{\mathrm{PSM}} - \log(2\pi){\cdot}\sum n_i$. This relation between the two objective functions was found to hold for all the estimated models on the four simulated data sets, and this demonstrates that the formulations of the objective functions are equivalent.

Applications

The general approach of including SDEs in the NLME model as implemented in PSM has a potential of improving model development and performance for a wide range of PK/PD modelling situations, as has been discussed previously. The applications to illustrate the functionality of PSM in the present paper have been chosen to focus on a feature inherent to the new approach. The SDEs enable a simple way to estimate unknown inputs and time-varying parameters by modelling these as a random walk. The technique works for both linear and non-linear problems, and this will be illustrated in the following by two models to estimate the insulin secretion rate and liver extraction rate.

Data

The data originates from a double-blind, placebo-controlled, randomized crossover study with a duration of 24 h starting at 8 a.m. in the morning. Thirteen patients (5 women and 8 men) with type II diabetes were examined. Their age given as mean $\pm$ 1 standard deviation was 56.4 ± 9.2 years, BMI was $31.2 \pm 3.6\,\mathrm{kg/m^2}$ and the duration of diabetes was 3.0 ± 2.6 years (range 5 months to 8 years) [15].

C-peptide and insulin measurements will be used for analysis in this paper, and only the placebo data is used. This is done to focus the presented analysis on two types of application of the NLME model which are only possible by extending it with SDEs, namely stochastic deconvolution of an unknown input and continuous tracking of the behavior of a parameter.

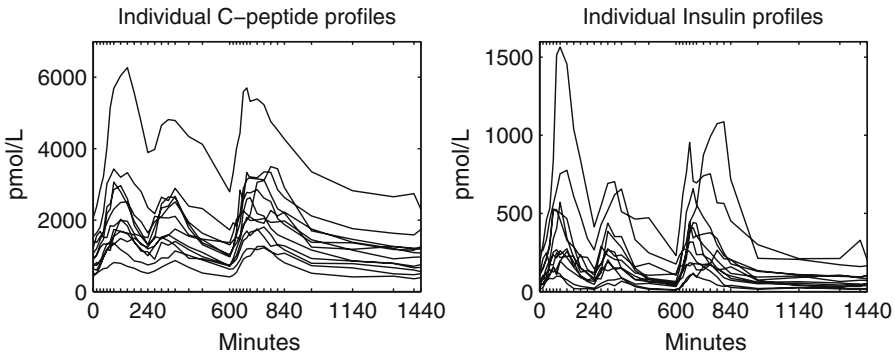


Fig. 1 Individual profiles for C-peptide and Insulin

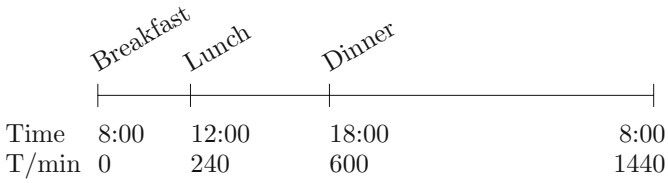


Fig. 2 Meal times during 24 h study period

One of the patients was discarded since the measurement times were delayed compared to the rest. The data used thus consists of 24-h C-peptide and insulin profiles for 12 individuals, see Fig. 1.

The subjects were sampled 35 times during the 24 h at varying time intervals, mainly concentrated after meal times. A total of 3 standard meals were given at 8 a.m., noon and 6 p.m., each to be finished within 20 min. These times correspond to 0, 240 and 600 min after the study was initiated, see Fig. 2.

Deconvolution

The first example of application will illustrate how the model setup can be used for deconvolution of the insulin secretion rate (ISR) based on a standard two-compartment linear model for C-peptide measurements [16]. It is known that C-peptide and insulin are secreted in equi-molar amounts, and this fact is used to construct the model. The basic idea is to model the secretion rate into the central compartment as a pure random walk (Wiener process) and then estimate ISR as the realization of this random walk using the EKF to provide a smoothed estimate.

The modelling of the ISR as a random walk actually means that no model is given for the ISR, and therefore it is instead estimated entirely based on the data. For a linear system this technique resembles a deconvolution, but it will provide a more smooth estimate compared to an ordinary deconvolution. This is because the EKF separates system noise from measurement noise, where the system noise for this model is assumed to be the ISR of interest. The extent of smoothing is determined by the maximum likelihood estimated σ_{ISR}, the magnitude parameter for the random walk

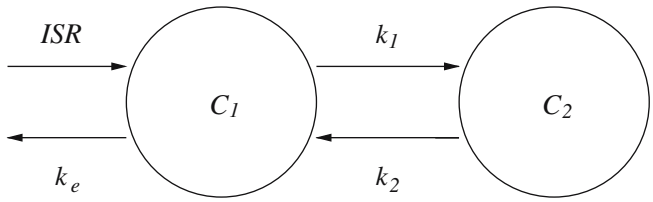


Fig. 3 Two-compartment model used for estimation of ISR

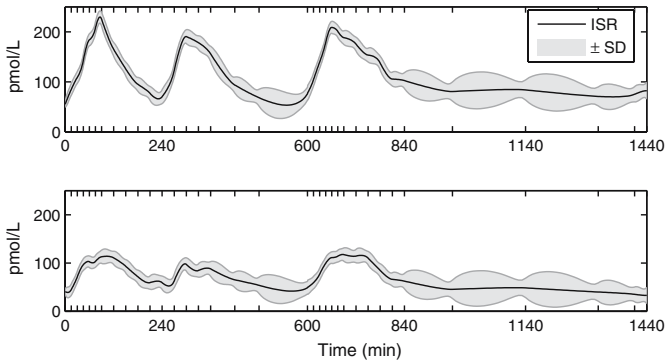


Fig. 4 Smoothed estimate of ISR for individual 1 and 2

for ISR, which influences the Kalman gain on increments of the random walk. A larger magnitude leads to a more fluctuating random walk with larger increments and vice versa for a smaller magnitude. The resulting estimate of the random walk and thereby the ISR is thus optimal in a likelihood sense, since the EKF as mentioned earlier has been shown to yield the minimum variance state estimate for a linear system.

The deconvolution setup requires three states, namely a central compartment state C_1 modelling the measured C-peptide concentration, a peripheral compartment state C_2, and a state ISR for the random walk. This gives the state vector $\boldsymbol{x} = [\, C_1\, C_2\, \mathrm{ISR}\,]^T$. The C-peptide kinetic parameters k_1, k_2, k_e are set equal to the Van Cauter estimates found in [17].

The C-peptide measurement error is assumed to be additive Gaussian white noise with variance Σ. The model states are constrained to steady state at $t = 0$ given an initial individually estimated concentration C_i in C_1, that is $\boldsymbol{x}_0 = \left[C_i \;\; \frac{k1}{k2}C_i \;\; k_e C_i\right]^T$ and $C_i = C_1^0 \exp\eta, \quad \eta \in N(0, \Omega_{C_1})$. The state equation for the model is shown in Eq. 18

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1 + k_e) & k_2 & 1 \\ k_1 & -k_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{x}\, dt + \mathrm{diag} \begin{bmatrix} 0 \\ 0 \\ \sigma_{\mathrm{ISR}} \end{bmatrix} d\boldsymbol{\omega} \tag{18}$$

and the measurement equation is simply $y = C_1 + \epsilon$, where $\epsilon \in N(0, \Sigma)$. The ML estimated population parameters are C_1^0, Σ, σ_{ISR} and $\Omega_{C_1^0}$ and based on these an optimal estimate of ISR can be found by using the Kalman smoothing algorithm.

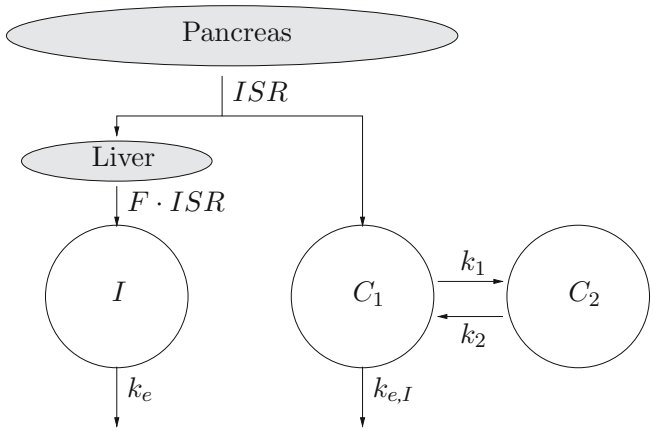


Fig. 5 Dynamics of the combined model for estimation of insulin secretion rate and liver extraction rate

Figure 4 shows the smoothed estimate of ISR for the first two individuals together with a ± 1 standard deviation band. The assumption of steady state in the beginning defines the initial level of ISR based on C_1^0 and this appears appropriate.

State-estimation

The second example of application goes to illustrate how the model setup may be used for state-estimation in non-linear systems. The method is also sometimes referred to as parameter tracking, when the state represents a parameter, which is suspected of having some time-varying behavior. Although non-linear state-estimation is fundamentally different from deconvolution, which only applies to linear systems, it can be performed with SDEs in basically the same way as the approach for deconvolution presented in the first example of application.

The aim is to estimate the dynamic liver extraction rate, which represents the fraction of insulin that is absorbed by the liver. This fraction is often modelled as a constant to simplify statistical models, although it is known to be time-varying. As previously done the insulin secretion rate is estimated based on the information in the C-peptide measurements and then used as input into a one-compartment insulin model. The state I models the measured insulin concentration in the compartment and the insulin elimination is set to $k_{e,I} = 0.355$ min^{-1}. This value has been reported for a similar study, also on type II diabetic patients [18]. By having a fixed elimination rate and ISR given from the C-peptide part of the data, the information in the insulin measurement can be used to estimate the liver extraction. The fraction which passes through the liver is modelled by a state F, and the input into the insulin compartment is thus $F \cdot$ ISR making the model non-linear in the states. The final layout of the model is shown in Fig. 5. The layout is identical to the layout first proposed in [19], where it is shown that by assuming a constant liver extraction rate it is possible to estimate the kinetic parameters and a piecewise constant ISR.

In an initial model F was modelled directly as a random walk in the same way as ISR. The estimation of the model returned a very low estimate of the insulin measurement

standard deviation at only 0.01 pmol/l. This is an unrealistically small value and indicates a problem with separation of noise components, since virtually all the variation in the insulin measurements thereby is assumed to originate from the fluctuations of the liver extraction.

This problem can be solved by imposing further smoothing to the state-estimation by choosing to model *the derivative* of F as a random walk instead of directly F as before. This is achieved by introducing a new state named X as shown in Eq. 19 and 20

$$\frac{dF}{dt} = X \tag{19}$$

$$dX = \sigma_X d\omega \tag{20}$$

where ω is a Wiener process. The change in the model for F causes the increments of the derivative of F to be penalized by the Wiener noise gain σ instead of the increments of F directly. The result is a less flexible model for F where fluctuations are further constrained, and the modification is easily implemented using the flexibility made available by the stochastic state space approach. In total the model contains six states, namely $\boldsymbol{x} = [C_1\ C_2\ I\ \mathrm{ISR}\ F\ X]^T$, which are all estimated simultaneously by the Extended Kalman Filter using the two-dimensional measurements with C-peptide and insulin. The system equations are shown in Eq. 21.

$$d\boldsymbol{x} = \begin{bmatrix} -(k_1 + k_e)C_1 + k_2 C_2 + ISR \\ k_1 C_1 - k_2 C_2 \\ -k_{e,I} I + F \cdot ISR \\ 0 \\ X \\ 0 \end{bmatrix} dt + \mathrm{diag} \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sigma_{\mathrm{ISR}} \\ 0 \\ \sigma_X \end{bmatrix} d\boldsymbol{\omega} \tag{21}$$

The estimation of the population parameters in the new model with a constrained model for F results in a better separation of noise. The standard deviation for the insulin secretion rate is estimated at a satisfactory level of 19.8pmol/l.

As could be expected, the model finds an ISR which is almost identical to the one found using just a C-peptide deconvolution model, since the information in the added insulin measurements is used to estimate the liver extraction. The smoothed estimate of the fraction of insulin passing the liver F is shown in Fig. 6 for individual 1 and 2. The plots illustrate that the proportion sent through the liver, F, is below one for the entire time interval as it naturally should be. This also holds for 8 out of the remaining 10 individuals. For the two last individuals F varies between 0.5 and 1.8. This is however not of great concern, because F and $k_{e,I}$ are correlated and it is thus probably just indicating that $k_{e,I}$ for this particular individual is set too high.

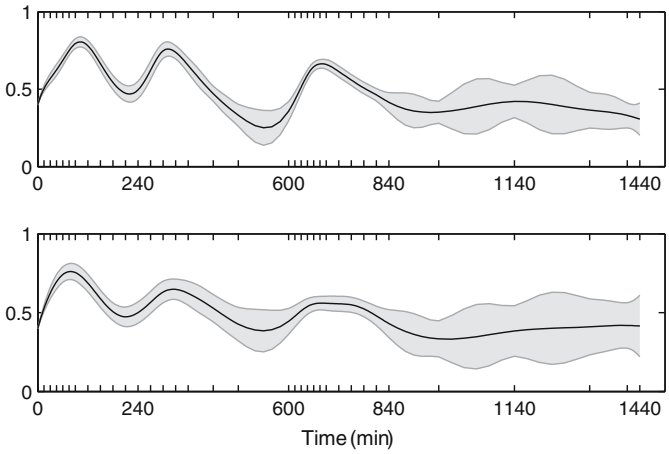


Fig. 6 Smoothed fraction of insulin passing the liver ±1 SD for individual 1 and 2

Discussion

By the presented software implementation PSM, we have shown that it is possible to develop a general purpose PK/PD population modelling tool that is able to handle the extra functionality made available by using SDEs in NLME models. The implementation opens up for the possibility to easily make further experiments with the model setup to allow for accumulation of more knowledge about the modelling approach.

It is important to emphasize that the software implementation is to be considered a prototype, which should only be used on research level. A necessary step to make it more widely usable is to move to another programming language. This implementation has been done in Matlab, which is ideal for numerical implementations, but it lacks in speed and parallel computing options. The standard within scientific programming today is Fortran, and this is also an obvious choice here due to its efficient handling of numerical computations and linear algebra calculations. Another advantage of Fortran is the accessibility of modules already available, such as algorithms for numerical optimization and ODE solvers.

The optimal platform for a future implementation is a shared memory system. Shared memory parallelism can be implemented easily in Fortran using the OpenMP[3] application program interface. OpenMP supports multi-platform shared-memory parallel programming in Fortran on all architectures, including Unix and Windows platforms. OpenMP is a scalable model that gives a flexible interface for developing a parallel application for platforms ranging from the desktop to the supercomputer and it supports parallelism through meta tags that will make portability to single CPU, multi-core CPU, and shared-memory multiprocessor (SMP) units simple. Some compilers are also able to create parallel calculations by automatically analyzing the code, but the largest improvements are achieved using manual parallelization.

[3] Further details may be found at www.openmp.org.

The present paper has illustrated how parallelization introduced at the individual minimizations of the population likelihood function has a strong potential of reducing the estimation time for a future final software program when dealing with data containing a large number of individuals. It can also be argued to introduce parallelism at an even higher level in the gradient calculation of the population likelihood function. This would generally be advantageous for models where the number of population parameters exceeds the number of individuals in data.

The first example of application in this paper demonstrated how the NLME model can be used for deconvolution of ISR by introducing SDEs. Although the estimation of ISR using SDEs is loosely denoted deconvolution, it is in fact not strictly speaking deconvolution but instead a probabilistic description of an unknown input, which is modelled as the realization of a stochastic process. Pure deterministic deconvolution using ODEs for the model shown in Fig. 3 will estimate ISR at each measurement to be equal to the rate giving the 'missing' amount in the central C-peptide compartment C_1. With the SDE approach the measurement noise on C-peptide is taken into account by the Kalman filter, which yields a minimal variance estimate of the states resulting in a more smooth estimate of ISR where the effect of noise is reduced.

Deconvolution based on noisy data is generally an ill-posed problem, meaning that even small perturbations in data lead to significant changes in the estimated solution [20]. The problem has been addressed by existing software by applying various kinds of regularization techniques to constrain the solution. An example is WinNonlin [21], which is a standard PK/PD software solution that can also be used for deconvolution. The program addresses the problem of deconvolution by introducing a smoothness factor and as a consequence it is simply left up to personal choice and preference of the user to specify the level of smoothing. An improved solution can be found using WinStoDec presented by [22], which is based on stochastic deconvolution and can be used for linear time-invariant systems [23]. It has been shown by [24] that the stochastic deconvolution approach is equivalent to the SDE approach presented here, which is furthermore by nature also able to handle non-linear time-varying systems, as has been demonstrated with the state-estimation approach in the second example of application presented here.

In conclusion, a fully functional prototype tool named PSM for estimation of NLME models based on SDEs has been implemented in Matlab and validated. The use of parallelization in the implementation has demonstrated a strong potential of reducing computation times in future implementations in a faster programming language. Finally two examples of application concerning insulin modelling demonstrated the possibility for deconvolution and non-linear parameter tracking facilitated by the extension of the NLME model to use SDEs.

Acknowledgments

The Authors wish to thank Ole Schmitz MD and his co-workers at The Institute of Pharmacology, University of Aarhus, for conducting the present study enabling us to perform the presented analysis.

Appendix: Approximation of population likelihood function

The population likelihood function for the NLME model with SDE's is defined in Eq. 13 as

$$L(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_{\omega}, \boldsymbol{\Omega}) = \prod_{i=1}^{N} \int \exp(l_i) d\boldsymbol{\eta}_i \tag{22}$$

where l_i is the individual a posteriori log-likelihood function. In most cases the integral cannot be evaluated analytically. For a general evaluation the individual a posteriori likelihood function can be approximated by a second order Taylor series expansion of l_i around the value $\hat{\boldsymbol{\eta}}_i$ of the individual random effects parameter which maximizes l_i. It follows that

$$l_i \approx l_i + \nabla^T l_i(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i) + \frac{1}{2}(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)^T \boldsymbol{\Delta} l_i (\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i) \tag{23}$$

$$\approx l_i + \frac{1}{2}(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)^T \boldsymbol{\Delta} l_i (\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i) \tag{24}$$

since $\nabla l_i = 0$ at $\hat{\boldsymbol{\eta}}_i$. Based on the approximation in Eq. 24 the integral in Eq. 22 can now be evaluated by moving constants such that the integral is over a multi-variate Gaussian density with mean $\hat{\boldsymbol{\eta}}_i$ and variance $(-\boldsymbol{\Delta} l_i)^{-1}$. This integral is equal to one and the result is

$$\int L_i d\boldsymbol{\eta}_i \approx \int L_i \cdot \exp\left(-\frac{1}{2}(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)^T(-\boldsymbol{\Delta} l_i)(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)\right) d\boldsymbol{\eta}_i \tag{25}$$

$$\approx L_i \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i}\right|^{\frac{1}{2}} \int \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)^T(-\boldsymbol{\Delta} l_i)(\boldsymbol{\eta}_i - \hat{\boldsymbol{\eta}}_i)\right) d\boldsymbol{\eta}_i \tag{26}$$

$$\approx L_i \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i}\right|^{\frac{1}{2}} \cdot 1 \tag{27}$$

$$\approx L_i \left|\frac{-\boldsymbol{\Delta} l_i}{2\pi}\right|^{-\frac{1}{2}} \tag{28}$$

where $L_i = \exp(l_i)$. The step in Eq. 28 is done to avoid a matrix inversion of the Hessian. By combining Eq. 22 Eq. 28 the population log-likelihood function can now be approximated by

$$L(\boldsymbol{\theta}, \boldsymbol{\Sigma}, \boldsymbol{\sigma}_{\omega}, \boldsymbol{\Omega}) \approx \prod_{i=1}^{N} \left|\frac{-\boldsymbol{\Delta} l_i}{2\pi}\right|^{-\frac{1}{2}} \exp(l_i)\Big|_{\hat{\boldsymbol{\eta}}_i} . \tag{29}$$

References

1. Aarons L (1999) Editorial–pharmacokinetic and pharmacodynamic modelling in drug development. Stat Methods Med Res 8(3): 181–182
2. Karlsson MO, Jonsson EN, Wiltse CG, Wade JR (1988) Assumption testing in population pharmacokinetic models: Illustrated with an analysis of moxonidine data from congestive heart failure patients. J Pharmacokinet Biopharm 26(2): 207–246
3. Karlsson MO, Beal SL, Sheiner LB (1995) Three new residual error models for population pk/pd analyses. J Pharmacokinet Pharmacodyn 23(6): 651–672
4. Kristensen NR, Madsen H, Ingwersen SH (2005) Using stochastic differential equations for pk/pd model development. J Pharmacokinet Pharmacodyn 32:109–141
5. Kristensen NR, Madsen H, Jørgensen SB (2004) Parameter estimation in stochastic grey-box models. Automatica 40: 225–237
6. Overgaard RV, Jonsson N, Tornøe CW, Madsen H (2005) Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm. J of Pharmacokinet Pharmacodyn 32(1): 85–107
7. Tornøe CW, Overgaard RV, Agersø H, Nielsen HA, Madsen H, Johnson EN (2005) Stochastic differential equations in nonmem®: Implementation application and comparison with ordinary differential equations. Pharm Res 22(8): 1247–1258
8. Beal SL, Sheiner LB (2004) NONMEM®users guide. University of California, NONMEM Project Group
9. Kalman RE, Bucy RS (1961) New results in linear filtering and prediction theory. Trans ASME Ser D J Basic Engrg 83: 95–108
10. Holst J, Lindström E, Madsen H, Nielsen HA (2000) Model validation in non-linear continuous-discrete grey-box models. Proceedings of the 13th IFAC symposium on system identification Rotterdam The Netherlands
11. Kristensen NR, Madsen H (2003) Continuous time stochastic modelling-ctsm 2.3 mathematics guide. Technical report Technical University of Denmark December
12. Gelb A, Kasper JF Jr, Nash RA Jr, Price CF, Sutherland AA Jr (1982) Applied Optimal Estimation. The MIT Press seventh edition
13. Nielsen HB (2000) Ucminf–an algorithm for unconstrained nonlinear optimization. Technical report IMM DTU
14. Dennis JE Jr, Schnabel RB (1983) Numerical methods for unconstrained optimization and nonlinear equations. Prentice Hall Series in Computational Mathematics, Prentice Hall Inc.
15. Degn KB, Juhl CB, Sturis J, Jakobsen G, Brock B, Chandramouli V, Rungby J, Landau BR, Schmitz O (2004) One week's treatment With the long-acting glucagon-Like peptide 1 derivative liraglutide (NN2211) markedly improves 24-h glycemia and alpha- and beta-Cell function and reduces endogenous glucose release in patients with type 2 diabetes. Diabetes 53(5): 1187–1194
16. Gabrielsson J, Weiner D (1997) Pharmacokinetic and pharmacodynamic data analysis: concepts and applications. Kristianstads Boktryckeri second edition
17. Van Cauter E, Mestrez F, Sturis J, Polonsky KS (1992) Estimation of insulin secretion rates from c-peptide levels. comparison of individual and standard kinetic parameters for C-peptide clearance. Diabetes 41(3):368–377
18. Kjems LL, Vølund A, Madsbad S (2001) Quantification of beta-cell function during ivgtt in type ii and non-diabetic subjects: assessment of insulin secretion by mathematical methods. Diabetologia 44: 1339–1348
19. Watanabe RM, Steil GM, Bergman RN (1998) Critical evaluation of the combined model approach for estimation of prehepatic insulin secretion. Am J Physiol 274(1): 172–183
20. Hadamard J (1923) Lectures on the Cauchy problem in linear partial differential equations. Yale University Press New Haven
21. Pharsight (2004) WinNonlin user manual version 3.1. Pharsight deconvolution edition
22. Sparacino G, Pillonetto G, Capello M, Nicolao G , De Cobelli C (2002) Winstodec: a stochastic deconvolution interactive program for physiological and pharmacokinetic systems. Comput Methods Programs Biomed 67: 67–77

23. de Nicolao G, Sparacino G, Cobelli C (1997) Nonparametric input estimation in physiological systems: problems methods and case studies. Automatica 33(5): 851–870
24. Kristensen NR, Madsen H, Ingwersen SH (2004) A deconvolution method for linear and nonlinear systems based on stochastic differential equations. poster presented at: population approach group in Europe (PAGE) 13th meeting June

Appendix B

Paper B

Title:
Introduction to PK/PD modelling with focus on PK and stochastic differential equations.

Authors:
S. B. Mortensen, A. H. Jónsdóttír, S. Klim, and H. Madsen.

Published in:
IMM-Technical Report-2008-16 (2008).

Introduction to PK/PD modelling

with focus on PK and stochastic differential equations

Stig Mortensen, Anna Helga Jónsdóttir,
Søren Klim and Henrik Madsen

June 23, 2009

DTU Informatics

DTU Informatics
Department of Informatics and Mathematical Modeling
Technical University of Denmark

Richard Petersens Plads
DTU - building 321
DK-2800 Kgs. Lyngby
Denmark

ISSN: 0601-2321

Contents

1 Introduction **2**

2 The ADME Model **3**

3 Fundamental concepts **4**

4 Compartment models **5**
4.1 One-compartment models 7
4.1.1 Case study: pain reliever 8
4.1.2 Flip-flop situation 10
4.1.3 Maximum concentration 11
4.1.4 Half-life of drug 11
4.1.5 Constant rate infusion and multiple dosing 12
4.2 Multi-compartment models 13
4.2.1 Case study: pain reliever continued 15

5 PD modelling **15**
5.1 Receptor Theory 16
5.1.1 Michaelis-Menten model 17
5.1.2 Commonly used PD models 18
5.2 Modelling with effect compartments 19
5.2.1 Case study: pain reliever continued 20

6 Modelling data **22**
6.1 Single individual 22
6.1.1 Error model using ODEs 22
6.1.2 Error model using SDEs 25
6.1.3 Case study: Advantages of using SDEs 27
6.1.4 Discussion of SDEs 30
6.2 Multiple individuals 31
6.3 Estimation 32
6.3.1 NONMEM 32
6.3.2 CTSM 33
6.3.3 PSM 33

A NLME log-likelihood function **33**

1 Introduction

The development of new medical drugs is driven by progress in many areas. These include medicine, biotechnology, new production equipment and not least, as will be the focus here, the area of mathematical and statistical modelling.

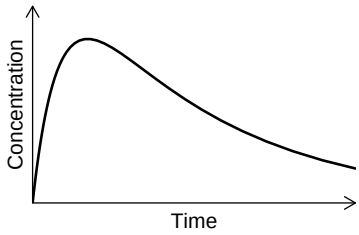
Before a new drug can move from a simple molecule in the laboratory to become a new product in the local pharmacy, there are many questions which must first be answered: Is it safe, also for patients, elderly people, pregnant women, etc.? Does the drug work? In which way should it be given to the patient? Are there any unwanted side-effects? The answer to all these questions require a long series of trials, which must be carefully planned to discover all facets of a new drug candidate.

New drug candidates are in the early development phase initially tested with animals. These tests aim to show if the drug seems to work and also to check for unwanted side effects. The next step is to move to clinical trials with humans. These trials have traditionally been separated into 4 phases. In Phase 1 the drug is given to healthy young male persons mainly to see if it is safe for humans but the sponsor (the drug company) will of course also look for indications of a positive effect. In Phase 2 the drug is given to the target patient group again mainly to show that it is still safe, but also to give indications of a positive effect. When safety has been established and the sponsor believes in the drugs potential, it will be moved to Phase 3, which generally involve the largest and most costly trials. These trials focus on proving the positive effect of the drug to the health authorities. If all goes well, the drug is approved and marketed. In some cases this will be followed by new trials, which is known as Phase 4 studies.

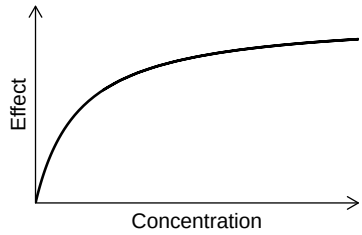
The series of clinical trials is not only very costly but also time consuming. The drug may easily take 10 years to get approved and marketed to the patients. Due to this both health authorities and drug companies are looking of ways to accelerate this project, while still keeping it as safe as possible. The primary target is to insure that no harmful drugs get approved, but at the same time that the good drug are not delayed unnecessarily from reaching the patients, who will benefit from them. It is in this connection that mathematical and statistical modelling has become an important tool, since it may help to give improved understanding of the outcome of clinical trials.

The scientific disciplines concerning mathematical modelling in drug development are called pharmacokinetics and pharmacodynamics, or in brief just PK/PD. In popular terms PK is often described as "what the body does to the drug" and PD as "what the drug does to the body". More specifically PK focuses on modelling how the drug passes through the body, normally by modelling concentrations in various areas of the body as a function of time, see Figure 1(a). PD aims at linking these modelled drug concentration to certain

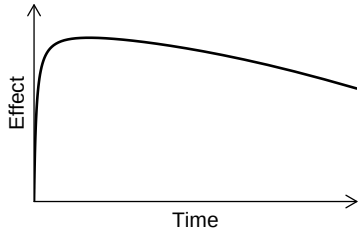
measure of effect through a PD-model. An example of a PD-model is shown in Figure 1(b). With a combined PK/PD model it is thus possible to give a picture of the expected effect for a given dose as a function of time as illustrated in Figure 1(c).



(a) PK model



(b) PD model



(c) Combined PK/PD model

Figure 1: Illustration of PK/PD modelling.

For more thorough reading on pharmacokinetics and pharmacodynamics the books Gabrielsson and Weiner (1997) and Rowland and Tozer (1997) are suggested.

2 The ADME Model

A complicated process is initiated as soon as a drug enters the body. This process can be divided into four phases, absorption, distribution, metabolism and elimination and hence the acronym ADME.

The absorption phase describes how the drug enters the body, or more precisely how the drug enters the bloodstream. When using intravenous (iv) administration, no absorption phase is present since the drug is injected directly into the bloodstream. The whole dose can be given in one rapid injection, called a bolus dose, or by using a constant rate infusion over a certain period of time. All other dosing methods, that is when the drug is not injected directly into

the bloodstream, are called extravascular dosing. Examples of such methods are injections into a muscle or fat tissue and oral dosing. Those methods have one thing in common, they require an absorption phase since the drug needs to cross some boundaries in the body before it reaches the bloodstream. As an example when administrating a pill (oral dosing), the pill needs to dissolve and cross the gut wall before it reaches the bloodstream.

The distribution phase describes how the drug spreads through the body, into its fluids and tissues, after it has reached the bloodstream. It is in the distribution phase the drug is brought to the place of action trough the bloodstream. The time it takes for the drug to get to the place of action is very dependent on if it is easily accessible by the bloodstream. The heart is, as an example, easily accessible by the bloodstream while the bone marrow is not.

The third phase, metabolism, describes a process where the initial (parent) compound is broken into another compounds, called metabolites. The metabolites can either be inactive, therefore reducing the drugs effect on the body, or they can be active, sometimes more active than the parent compound. The liver plays a leading role in metabolism since it produces many of the enzymes used by metabolism.

The last phase, the elimination phase, describes how the compounds and their metabolites are removed from the body via excretion. Most drugs are eliminated via the kidneys with urine.

The four phases of the ADME model can be summarized as:

Absorption Drug entering the body

Distribution Drug is spreading to different areas of the body

Metabolism Drug is being changed to new chemical compounds

Elimination Drug is removed from the body

3 Fundamental concepts

Concentration is defined as amount per volume and is the most central concept in PK/PD modelling. The reason is that concentration of a drug is relatively easy to measure from a blood sample and at the same time concentration is a key factor when modelling both positive and negative effects of a drug. Concentration is calculated as

$$C = \frac{A}{V} \tag{1}$$

where A is amount of drug and V is the volume of distribution. The amount of drug is either measured as mass (mg) or in number of molecules (mol). The volume of distribution is defined as the volume which the drug has to distribute evenly into in order to reach the measured concentration in the blood, C. If the the drug only distributes into the blood then the volume of distribution will be equal to the volume of the blood. However, often the volume of distribution will be larger than the volume of the blood. This can happen if the drug distributes into other parts of the body or if the drug is chemically bound in a way where it cannot be measured. The amount of drug is unchanged, but only a smaller part can be measured in the blood. This will result in a larger volume of distribution to reflect the lower measured concentration of the drug.

Another central issue is the timecourse of the elimination of the drug. In the most simple model one assumes that the elimination rate is proportional to the remaining amount of drug, A. The proportionality constant is CL/V, where CL is called clearance and measures the volume of blood which is cleared for drug per time. The rate of clearance is thus $CL/V \cdot A$. CL and V are sometimes referred to as micro constants and are sometimes replaced by

$$K = \frac{CL}{V} \ , \tag{2}$$

where K, the elimination rate constant, is a so called a macro constant. The choice of parameterization will be decided by the identifiability of the parameters based on the given data. In some situations it is possible to estimate both CL and V but some times they cannot be separated and it is thus only possible to estimate K.

A basic measure of the exposure of the drug is called AUC, which stands for area under the curve. AUC measures the area under the curve of concentration vs. time. An important feature of AUC is that it can be evaluated for most types of models and it can even be determined graphically based of a series of concentration measurements.

4 Compartment models

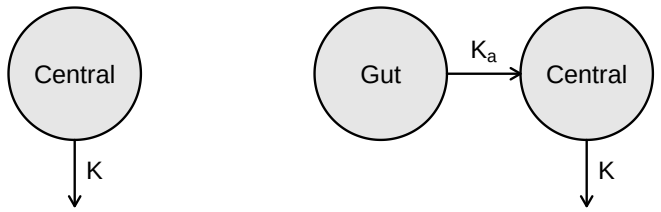
The purpose of PK modelling is mainly to describe how a drug passes through the body by modelling concentrations of the drug in different areas of the body. To measure the concentration of a drug a blood sample is usually taken and the concentration measured. Since the heart is pumping blood constantly it can be assumed that the concentration of the drug is the same within the bloodstream at a given time. This means that as soon as the drug has reached the bloodstream the concentration of the drug is the same throughout the bloodstream. However, it might not be the case that the drug spreads instantly to other parts

of the body. Therefore, in order to build mathematical models to describe how the concentration changes with time the body can conveniently be divided into parts, called compartments, where the drug can be assumed to behave in the same manner. This type of modelling is called compartment modelling. The compartment where the concentration is measured, usually the bloodstream, is of special interest and is called the central compartment.

The most basic model found is the one describing a one compartment system, which only includes the central compartment and a possible absorption compartment. This model is appropriate to use if the drug distributes to accessible areas of the body instantly. If the drug is given directly into the bloodstream (iv) the system only includes a central compartment while in the case of extravascular dosing the system should have an absorption compartment in addition to the central compartment. As an example, if the drug is administrated orally the system should include a gut compartment.

In some cases the one compartment model is not suitable for describing the system and in these cases a multi compartment model may need to be applied (see Sec. 4.2). However, in the remaining part of this report the focus will mainly be on the one compartment model (with an additional compartment if the drug is not administrated directly into the bloodstream) since many systems can be described using that model.

A compartment system can easily be visualized by drawing the compartments as circles and the connections between them with arrows indicating the direction of the flow between the compartments. A figure showing the one compartment system in case of intravenous bolus dose is shown in Figure 2(a) and in the case of oral dosing in Figure 2(b).



(a) 1-compartment system

(b) 1-compartment system with additional gut compartment

Figure 2: Illustration of compartment models.

The transfer rate of a drug from one compartment to another can usually be described using first order kinetics, meaning that the rate of change is proportional to the amount of the drug in the source compartment.

4.1 One-compartment models

The relationship between the rate of elimination and the amount of a drug in a one compartment model with first order kinetics when drug is administrated as a bolus dose (the system shown in Figure 2(a)) can be written mathematically as:

$$\frac{dA}{dt} = -K \cdot A \tag{3}$$

where A is the amount of the drug and K is the first order elimination rate constant. K is always positive and its size controls the speed of the elimination. The differential equation can be solved resulting in a function describing the amount of the drug in the central compartment at a given time

$$A_{bolus}(t) = A_0 \exp(-K \cdot t) \tag{4}$$

where A_0 is the amount of the drug at time $t = 0$, that is the size of the given dose.

In the case of extravascular dosing other compartments needs to be added to the model. E.g. for oral dosing a extra gut compartment is often sufficient to model the absorption phase (the system shown in Figure 2(b)). Usually the rate of change in the gut compartment can be described with first order kinetics resulting in the following differential equation

$$\frac{dA_{gut}}{dt} = -K_a \cdot A_{gut} \tag{5}$$

where A_{gut} is the amount of the drug in the gut and K_a is the first order absorption constant. It is usually the case that $K_a > K$ meaning the absorption of the drug from the gut into the central compartment is faster than the elimination process. However, in some cases $K_a < K$ which is known as the flip-flop situation. The flip-flop situation is discussed further in Section 4.1.2. The change in amount in the central compartment can now be found by combining (3) describing the elimination, and (5) describing the absorption from the gut resulting in

$$\frac{dA}{dt} = \overbrace{F \cdot K_a \cdot A_{gut}}^{\text{from gut}} - \overbrace{K \cdot A}^{\text{elimination}} \tag{6}$$

where F denotes the bioavailability which is the fraction of the dose that reaches the central compartment. The differential equation can be solved resulting in an expression for the amount of drug in the central compartment for a given time which is a function of both the absorption and the elimination:

$$A_{oral}(t) = \frac{K_a F A_0}{K_a - K} \left(\exp(-K \cdot t) - \exp(-K_a \cdot t)\right) \tag{7}$$

Equations (4) and (7) can now be used to describe the amount of drug in the central compartment for a one compartment system in the case of a bolus dose and oral dosing, respectively. Usually it is more interesting to model the concentration (C) in stead of the amount since it is the concentration of the drug in the blood that is measured. According to (1) the concentration is found by dividing the amount by the volume of distribution resulting in

$$C_{bolus}(t) = \frac{A_{bolus}(t)}{V} = \frac{A_0}{V} \cdot \exp(-K \cdot t) \tag{8}$$

in the case of a bolus dose, and

$$C_{oral}(t) = \frac{A_{oral}(t)}{V} = \frac{K_a F A_0}{V(K_a - K)} \left(\exp(-K \cdot t) - \exp(-K_a \cdot t)\right) \tag{9}$$

in the case of oral dosing.

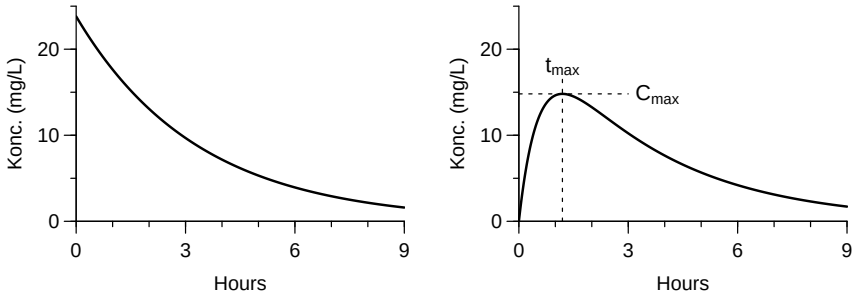
4.1.1 Case study: pain reliever

To illustrate the use of compartment modelling the drug paracetamol will be used as a case study. Paracetamol is the active substrate in a large number of pain relieving drugs on the market although its mechanism of action is still a source of debate. At correct dosages it works well against head ache and fever but at very high dosages it can cause lasting damages on the liver. For adults it is recommended to take doses of 1000mg at most 3-4 times a day and never more than 4g per day.

Paracetamol has been tested in a number of trails and it has been shown that the pharmacokinetics can be adequately described by a multi-compartment model structure, which will be introduced in Section 4.2 in further detail (see also Rawlins et al. (1977)). As a good approximation however, it can be modelled using a one-compartment model with a 1st order elimination from the blood and likewise a 1st order absorption from the stomach. Paracetamol can thus be modelled with the systems shown in Figure 2 for intravenous bolus and oral dosing. Based on Rawlins et al. (1977) the elimination rate constant can be found to $K = 0.28h^{-1}$ and the volume of distribution is $0.60L/kg$ giving $V = 42L$ for an average $70kg$ adult. The absorption is controlled by an absorption rate constant of $K_a = 1.80h^{-1}$ and the bioavailability is found to $F = 0.89$. Note that $K_a > K$ holds here, meaning that paracetamol is absorbed faster than it is eliminated.

Based on the information found for paracetamol it is possible to draw the concentration as a function of time based on (8) and (9). This is called a

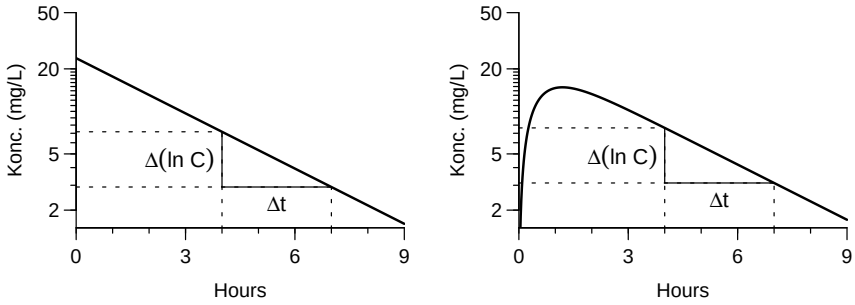
concentration profile and is shown for both intravenous bolus and oral dosing in Figure 3.



(a) Intravenous (iv) bolus dose (b) Oral dose

Figure 3: Concentration profiles for dosing of $1000mg$ paracetamol.

Concentration profiles are also commonly shown with concentration on a log-scale as seen in Figure 4. This makes it possible to directly read off the elimination rate constant K as the slope of the line for intravenous bolus dosing. For oral dosing K is found as the slope of the last part of the profile (terminal slope) where it follows a straight line. This holds since $K_a > K$ and the absorption from the gut thus has finished so the drug is only contained in the blood as for intravenous bolus dosing.



(a) Intravenous (iv) bolus dose (b) Oral dose

Figure 4: Concentration profile on log-scale for dosing of $1000mg$ paracetamol.

In this case looking at Figure 4(b) we find approximately

$$K = -\frac{\Delta(\log C)}{\Delta t} = -\frac{\log 3 - \log 7.5}{7h - 4h} \approx 0.305h^{-1} \tag{10}$$

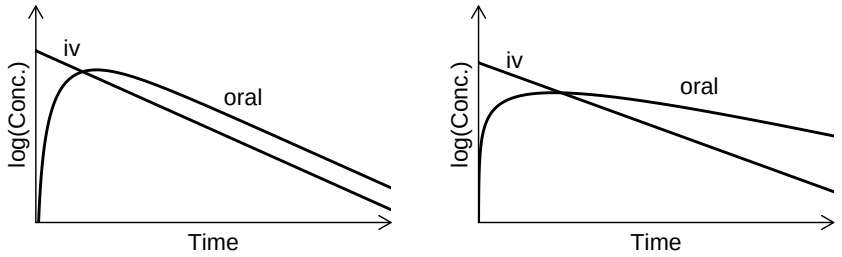
which compares well to the true value of $K = 0.28h^{-1}$.

4.1.2 Flip-flop situation

Cases where the absorption rate constant is larger than the elimination rate constant ($K_a < K$) is called a flip-flop situation. In these situations the absorption will be the so-called rate limiting step in the final phase of the elimination for oral dosing (Gabrielsson and Weiner 1997) and thus the terminal slope for oral dosing will be $-K_a$ instead of $-K$ as shown in Figure 4(b). For intravenous dosing the slope is $-K$ independent of K_a.

In order to be able to decide if it is a flip-flop situation it is necessary to perform both an intravenous bolus dosing study and an oral dosing study. If the terminal slope of the oral dosing concentration profile is parallel to the intravenous bolus concentration profile it is a normal situation, since the final elimination rates are equal. This can be known since the intravenous bolus concentration profile consists only of an elimination phase without an absorption phase.

On the other hand, if it is observed that the terminal slope for oral dosing is less steep than the intravenous bolus profile slope then the final phase must be absorption. The two situations are illustrated in Figure 5.



(a) Terminal phase for oral dosing is elimination

(b) Terminal phase for oral dosing is absorption

Figure 5: Situations with both elimination (normal) and absorption (flip-flop) as terminal phase seen as parallel and non-parallel terminal slopes respectively.

It is possible to determine the remaining rate constant (either K_a in a normal situation and K in a flip-flop situation) using the method of residuals. This is explained in detail in Gabrielsson and Weiner (1997).

4.1.3 Maximum concentration

It is often of importance to know the maximum concentration, C_{max}, of a drug in the blood and the time it takes to reach this maximum, t_{max}. For an intravenous bolus dose the maximum concentration is obtained just after the drug is injected into the the bloodstream, that is $t_{max} = 0$ and the maximum concentration can be calculated as

$$C_{max,iv} = \frac{A_0}{V} \tag{11}$$

where A_0 is the size of the dose and V is the volume of distribution.

In the case of extravascular administration, the concentration will not peak until after a while because of the absorption step. In general, the time it takes to reach the maximum can be found by differentiating the expression for $C(t)$ for the system, with respect to t, set the derivative equal to zero and finally solve for t_{max}. In the case of oral dosing in a one compartment system following first order absorption and elimination the expression for $C(t)$ is given in (9). Differentiating this expression, setting the derivative to zero and solving for t_{max} gives

$$t_{max} = \frac{1}{K_a - K} \ln\left(\frac{K_a}{K}\right) \tag{12}$$

The resulting maximum concentration at t_{max} then becomes

$$C_{max,oral} = \frac{K_a F A_0}{V(K_a - K)} \left(\exp(-K \cdot t_{max}) - \exp(-K_a \cdot t_{max})\right) \tag{13}$$

which can be simplified to

$$C_{max,oral} = \frac{F A_0}{V} \exp(-K \cdot t_{max}) \tag{14}$$

4.1.4 Half-life of drug

An important property of a drug is its biological half-life, $t_{1/2}$. The half-life is the time it takes for reducing the amount of drug left in the body by 50%. In the case of a bolus dose, in a one-compartment system, the amount of the drug in the body at $t = t_{1/2}$ is

$$A(t_{1/2}) = A_0 \exp(-K \cdot t_{1/2}) \tag{15}$$

according to (4). By definition, half of the given amount (A_0) should be left at in the body at $t = t_{1/2}$ or

$$\frac{1}{2} A_0 = A_0 \exp(-K \cdot t_{1/2}) \tag{16}$$

which can be simplified to

$$t_{1/2} = \frac{\ln 2}{K} \tag{17}$$

4.1.5 Constant rate infusion and multiple dosing

It is often the case that it is not enough for a patient to have effect of a drug during the time span where a single pill or intravenous bolus dose is active in the body. In some cases the solution is simply to give a higher dose, but since this may cause unwanted side effects, this is not always the best way to go.

Another possibility to prolong the effect of a drug is to give it as a constant rate infusion into the vein. This is the best way to control the drug flow into the body and it is easily modelled for a constant rate R_{in} by

$$\frac{dC}{dt} = \frac{R_{in}}{V} - \frac{CL}{V} \cdot C \tag{18}$$

assuming a one-compartment model with first order elimination. The solution is given as

$$C(t) = \frac{R_{in}}{CL}\left[1 - \exp\left(-\frac{CL}{V}t\right)\right] \ . \tag{19}$$

In some cases it is more practical to approximate the constant rate infusion by taking pills with a constant time interval. This is known as multiple dosing and is related to a constant rate infusion by the equation

$$R_{in} = \frac{F \cdot A_0}{\tau} \tag{20}$$

where A_0 is the dose in each pill, F is the bioavailability (the fraction that reaches the blood) and τ is the time interval between dosing. The concentration profile for multiple dosing is a sum of single oral dosing profiles which can be written as

$$C_{MD}(t) = \sum_{n=0}^{N-1} C_{oral}(t - n\tau) \tag{21}$$

where N is the number of doses and C_{oral} is given in (9).

The system is in steady state when the elimination rate equals the infusion rate, that is when

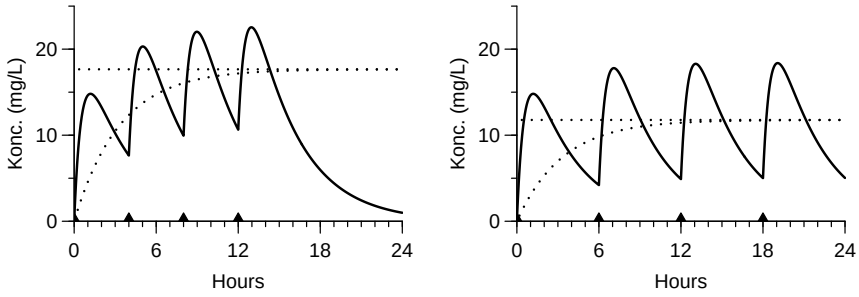
$$\frac{R_{in}}{V} = \frac{CL}{V} C_{SS} \tag{22}$$

which gives a steady state concentration

$$C_{SS} = \frac{R_{in}}{CL} \ . \tag{23}$$

It can be shown by using (19) that 90% of C_{SS} is reached after 3.32 half-lives. This result is independent of the rate of infusion which gives rise to the general rule of thumb that 90% of steady state concentration is reached after 3-4 half-lives (Gabrielsson and Weiner 1997).

Two examples of constant rate infusion and multiple dosing concentration profiles for paracetamol are shown in Figure 6.



(a) 4 hours between oral dosing

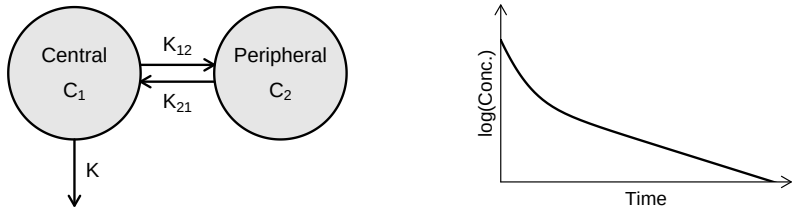
(b) 6 hours between oral dosing

Figure 6: Multiple dosing of 4g paracetamol with 4 oral doses of 1g shown as a thick line. The dotted line is a constant rate infusion at a corresponding rate.

4.2 Multi-compartment models

It may be the case that the one-compartment model is not sufficient to describe the distribution and elimination of a drug. More complicated models where additional compartments are added to the central compartment, resulting in a multi-compartment system, should then be applied. As a consequence, the system consists of a central compartment, representing the bloodstream and rapidly equilibrated organs, one or more peripheral compartments, representing more slowly equilibrating tissues, and finally in the case of extravascular administration, an absorption compartment.

The expression for $C(t)$ for a one compartment system (intravenous dosing) only includes a singe exponential term (Equation (4)). The best way to reveal how many compartments are needed to best describe the time course of the concentration is to plot the concentration on a semi-logarithmic scale. For a multi-compartment system this will most likely look like a piecewise linear function. As a general rule of thumb one compartment is needed for each linear part that is identified. As an example a two compartment system in the case of a bolus dose and the corresponding concentration profile are shown in Figure 7. It can be seen by looking at the figure that the concentration profile consists of two linear phases, a rapid initial and a slow terminal phase, which is the "fingerprint" of two-compartment systems.



(a) Two-compartment system

(b) Concentration profile for a two-compartment system

Figure 7: Illustration of two-compartment models.

The system shown in Figure 7(a) can be described mathematically using two differential equations, where C_1 and C_2 represents the concentration in the central compartment and the peripheral compartment respectively.

$$\frac{dC_1}{dt} = K_{21} \cdot C_2 - K_{12} \cdot C_1 - K \cdot C_1 \tag{24}$$

$$\frac{dC_2}{dt} = K_{12} \cdot C_1 - K_{21} \cdot C_2 \tag{25}$$

A solution of the differential equations, that is an expression for the concentration in the central compartment, can be written as

$$C = A \cdot \exp(-\alpha t) + B \cdot \exp(-\beta t) \tag{26}$$

Expressions for A, B, α, β given by K_{12}, K_{21} and K can be found in Gabrielsson and Weiner (1997). The half-lives of the two phases can finally be calculated as

$$t_{1/2,\alpha} = \frac{\log(2)}{\alpha} \tag{27}$$

and

$$t_{1/2,\beta} = \frac{\log(2)}{\beta} \ . \tag{28}$$

4.2.1 Case study: pain reliever continued

To illustrate the use of compartment modelling, the intake of paracetamol was modelled using a one compartment model in Section 4.1.1. It has however been shown that paracetamol concentration, after intravenous bolus dosis, follows a bi-exponential decline indicating that a two compartment model should be used to describe the system. Based on Rawlins et al. (1977), the time course of the concentration following an intravenous bolus dose of 1000mg is given by

$$C(t) = 13.8 \cdot \exp(-2.55t) + 13.0 \cdot \exp(-0.28t) \tag{29}$$

The resulting concentration profile is shown in Figure 8.

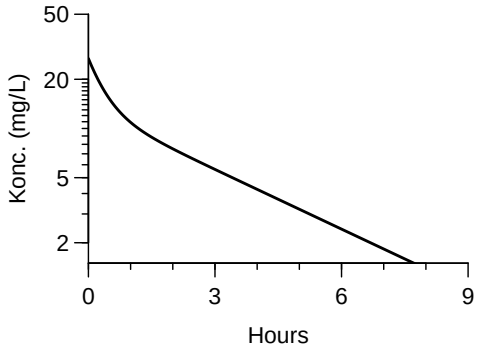


Figure 8: Concentration profile for paracetamol (1000mg intravenous bolus).

5 PD modelling

This section contains a brief introduction to pharmacodynamics (PD). In pharmacodynamics the dependent variable is not always straight forward to define. The dependent variable in pharmacokinetics is amounts or concentrations of the drug but for pharmacodynamics the dependent variable is not so obvious. How is the effect of a drug measured? Drugs against high blood pressure or fewer can be measured on the actual drop in pressure or temperature. Within epilepsy the desired effect of a drug should lower the number of seizures making the PD measurement a count. Within pain relieve the measurement can be a scale where the subject grades the pain and here the response variable will be an ordinal variable.

This wide variety of possible outcomes of a pharmacodynamic trial makes it hard to present a single methodology to handle all cases of PD modelling. This

section will focus on the most simple models and only with continuous response variables.

5.1 Receptor Theory

This section includes a quick introduction into receptor theory and how it helps in the understanding of the pharmacodynamic response. The presentation of the receptor theory is highly inspired by Gabrielsson and Weiner (1997).

Before a drug molecule can give rise to a pharmacodynamic effect it needs to interact with the cells. The cell membrane is covered in different receptors each with a specific structure. The structure defines which molecules that can attach to the receptor. In engineering terms a receptor can best be described as a docking station. The drug binding to the receptor initiates a change in the structure of the receptor and thereby changing the cell membrane.

Drugs are divided into two classes according to their function on the receptor. *Agonists* initiates a structural change in the receptor thereby changing the cell resulting in a response. *Antagonists* have a different role by simply binding to the receptor but not inducing any response. By occupying the receptor it blocks the receptor for other molecules. This is also why antagonists are sometimes referred to as blockers.

The set of unoccupied receptors $[R]$ placed on the target cells that potentially can bind with drug molecules $[D]$ and their relation to the bound drug/receptor complex $[DR]$ can be stated as

$$[D] + [R] \quad \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} \quad [DR] \tag{30}$$

where D denotes the drug, R the receptors and DR the bound drug/receptor complex.

The equation is based on a reversible receptor that can release the drug and be occupied by a new drug molecule. The constant k_1 indicates the rate of change from unbound to bounded and k_{-1} the other way. The binding property between the drug and the receptor determines the proportion of bound drug at equilibrium. The term affinity, which is often used to describe how good a drug binds to the receptor, is defined as

$$\text{Affinity} = \frac{k_1}{k_{-1}} = \frac{1}{K_d} \ . \tag{31}$$

The inverse of the affinity is denoted the disassociation constant K_d as shown above and is also often used.

The properties of association and disassociation determines the proportions in each state at equilibrium. It is clear that both the willingness to association/bind and the ability to stay associated/connected affects the proportions in equilibrium. One easy way to affect the amount of bound drug is simply to add more drug.

The link from the drug/receptor complex to the pharmacodynamic response can be thought of as a direct link between the receptor occupancy and the response. However often an extra state is included although hard to measure. The extra state extends the occupied state by introducing an occupied and activated state. It splits the assumption that an occupied receptor is automatically active.

$$\overbrace{\overbrace{[D]+[R]}^{\text{Inactive}} \rightleftarrows [DR]} \rightleftarrows \overbrace{[DR^*]}^{\text{Active complex}} \tag{32}$$

Equation (32) should be interpreted using the concepts of binding and activation. Now a drug can be specified as both having a property for binding and activation of the receptor. This ability to activate the receptor is denoted *intrinsic activity* and is more difficult to measure (Gabrielsson and Weiner 1997).

5.1.1 Michaelis-Menten model

It is possible to derive a model for the relation between drug concentration and effect based on receptor theory. The effect response E is assumed proportional to the occupancy of the receptors and thus that the maximal effect is achieved if all receptors $[R_{tot}]$ are occupied. This can be stated as

$$E = \alpha[RD] \tag{33}$$
$$E_{max} = \alpha[R_{tot}] \tag{34}$$

where $[R_{tot}] = [R] + [RD]$ is the total number of receptors and E_{max} is the maximal effect and α is the proportionality constant.

The steady state conditions can be stated as

$$\begin{aligned} \frac{d[RD]}{dt} &= k_1[R][D] - k_{-1}[RD] = 0 \\ \frac{[R][D]}{[RD]} &= \frac{k_{-1}}{k_1} = K_d \end{aligned} \tag{35}$$

By substituting $[R]$ with $([R_{tot}] - [RD])$

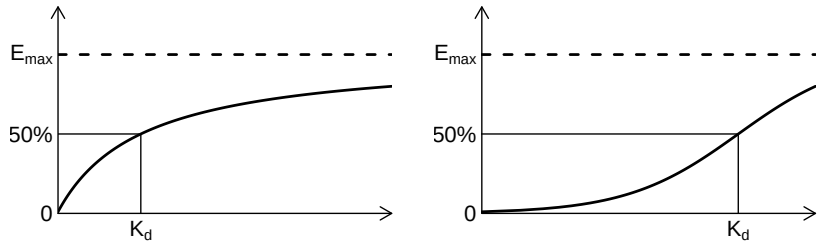
$$\begin{aligned} \frac{[D]([R_{tot}] - [RD])}{[RD]} &= K_d \\ \frac{[RD]}{[R_{tot}]} &= \frac{[D]}{[D] + K_d} \end{aligned} \tag{36}$$

Now by inserting the response assumptions

$$\begin{aligned} \frac{E/\alpha}{E_{max}/\alpha} &= \frac{[D]}{[D] + K_d} \\ \frac{E}{E_{max}} &= \frac{[D]}{[D] + K_d} \\ E &= \frac{E_{max}[D]}{[D] + K_d} \end{aligned} \tag{37}$$

The relationship in (37) is called a Michaelis-Menten relationship between drug concentration and the effect. This derivation demonstrates the motivation for the use of saturable models in pharmacodynamic modelling. Physiologically it also makes sense that at some point increasing the drug concentration will not result in a increased response.

The disassociation constant K_d determines the concentration at $1/2 \cdot E_{max}$ as can be seen from (37). An example of the model on both normal and logarithmic scale is shown in Figure 9.



(a) Response vs. Conc. (b) Response vs. log(Conc).

Figure 9: Michaelis-Menten relationship with response.

5.1.2 Commonly used PD models

The Michaelis-Menten relationship forms the basis of one of the most commonly used models to describe the relation between effect and concentration. This model is called the Sigmoid Emax model and is described by

$$E = E_0 + \frac{E_{max} C^n}{C^n + \mathrm{EC}_{50}^n} \tag{38}$$

where C is the concentration and EC_{50} is the concentration at $E_0 + 1/2 \cdot E_{max}$. The extra parameter n is included to provide a more flexible model. The model is often used with $n = 1$ and then simply called an Emax model.

In order to be able to estimate parameters in the (Sigmoid) Emax model it necessary to have estimates of the effect all the way from E_0 up to a point where maximum effect $E_0 + E_{max}$ seems to have been reached. If this is not the case it is often advisable to use a more simple model such a the linear model

$$E = E_0 + S \cdot C \tag{39}$$

or the log-linear model

$$E = m \cdot \log(C + C_0) \ . \tag{40}$$

Both of the models will in many situations be able to adequately describe the observed concentration-effect relationship. A comparison of all three models are shown in Figure 10. For the Emax model $n = 1$, $E_0 = 3$, $E_{max} = 8$ and $\mathrm{EC}_{50} = 100$ and remaining parameters S, m and C_0 are chosen so they all have the same effect at concentrations of 0 and EC_{50}.

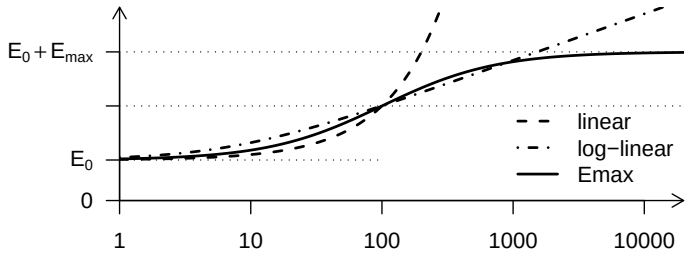


Figure 10: Comparison of standard models in PD analysis.

5.2 Modelling with effect compartments

In many situations it is not enough to directly model the response as a function of systemic concentrations in the PK model. This can happen if maximum effect is delayed compared to the maximum concentration.

This can numerically be handled by adding an additional compartment with concentration C_e representing the near cell tissue. This is called an effect compartment and is assumed to have a negligible volume. There will thus not be any mass transfer from the PK model. Furthermore a rate parameter governing the time delay from systemic concentration C_1 to near cell concentration is needed and for identifiability the same parameter is often used as elimination from the effect compartment. The model for the effect compartment is

$$dC_e/dt = k_{e1}C_1 - k_{e0}C_e \ . \tag{41}$$

The response from the drug is now modelled as a link from the effect compartment to the response. The link can be either a linear, log-linear, Michaelis-Menten or an even more complex relationship.

The response functions is often extended with the use of subject specific covariates as this can increase the accuracy of the model.

5.2.1 Case study: pain reliever continued

This example shows how a PD model can be build on top of a PK model to model the effect after an oral dose of 1000mg paracetamol. The PK-model used is the two-compartment model shown in Sec. 4.2.1. This is used in combination with a first order absorption from the gut. The PK model is defined below in (42) to (44).

$$\begin{aligned} dC_{gut}/dt &= -K_a C_{gut} && (42)\\ dC_1/dt &= -k_{12}C_1 + k_{21}C_2 - k_{10}C_1 + FK_aC_{gut} && (43)\\ dC_2/dt &= k_{12}C_1 - k_{21}C_2 && (44) \end{aligned}$$

The PD model used in this example is taken from (Gibb and Anderson 2008). It uses an effect compartment with an Emax model to model the effect. The effect compartment model is shown in (41) and is only a hypothetical compartment to introduce a delay of effect. It does thus not influence the PK model.

The compartment structure of the combined PK/PD model is shown in Figure 11. The arrows for the effect compartment are shown with dashed lines to indicate the there is no actual mass transfer.

The effect is measured on a visual analogue scale (VAS) from 0-10 where reduction below 10 indicates pain relief. The Emax model for the effect is

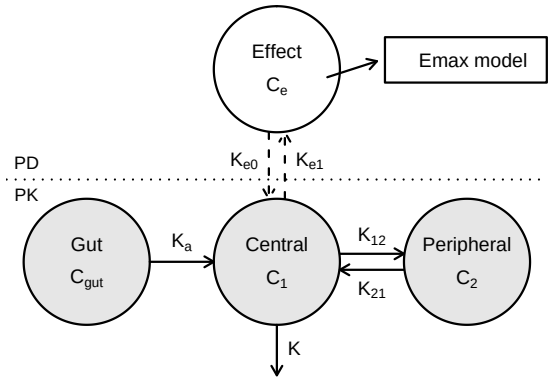
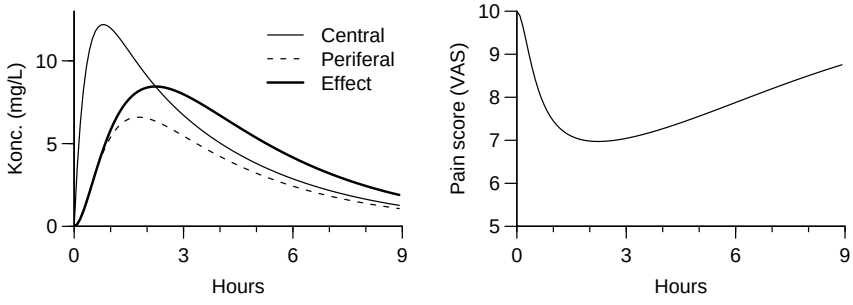


Figure 11: PK/PD model for paracetamol.

$$\text{Effect} = 10 - \frac{E_{max} C_e}{\text{EC}_{50} + C_e}$$

with $E_{max} = 5.17$ and $\text{EC}_{50} = 9.98$mg/L. The rate constants for the effect compartment are $k_{e0} = k_{e1} = 0.83h^{-1}$ giving a half-life of 50 min. A simulation of the combined PK/PD model is shown in Figure 12.

By looking at the model for the effect in Figure 12(b) it can be seen that pain relief following a 1000mg oral dose can be expected after 0.5-1 hour and it seems to last around 6 hours or more. With the combined PK/PD model it is now also possible to give estimates of the expected effect at e.g. half or double dose without actually doing the experiment, although results from extrapolation should always be treated with care.



(a) Concentrations after 1000mg oral dose.

(b) Model of pain relief.

Figure 12: Illustration of PK/PD model for paracetamol.

6 Modelling data

To work with real life data a model of a system must be able to handle noisy observations. The models discussed until now has only been concerned with relatively simple deterministic relationships, but real systems naturally contain much more variation than suggested by these simple models. The remaining part of the text will focus on how to extend these models to include a model for the different types of variation found in data. The aim is to enable the use of a reliable statistical framework for consistent inference, simulation, prediction and control by providing a proper description of the variation into the model.

The variation found in data can be caused by a number of sources. The main sources include ordinary measurement variation and variation due to difference between individuals, sites, occasions, etc. There may also be stochastic fluctuations of the system within an individual or approximations in the applied model, which will also lead to variation in data that must be accounted for. Further more, the model for an individual may depend on an input process (e.g. room temperature) which is sampled continuously together with the response. Measurements of this process is normally assumed to be done without measurement error, but if this is not true it will also lead to variation in the data that must be included in the model.

This section will discuss how to handle all of these different sources of variation.

6.1 Single individual

The structure of data for a single individual is

$$\boldsymbol{y}_j \ , \qquad j = 1...n \tag{45}$$

where $\boldsymbol{y}_j$ is a possibly multi-dimensional response. The sub-index is short hand notation referring to the sampling times $t_0 < t_j < t_n$.

6.1.1 Error model using ODEs

Modelling of PK for single individuals has traditionally been based on ordinary differential equations (ODEs) as for example done in a classical tool like NONMEM (Beal and Sheiner 2004). The observed deviations from the deterministic part of this model is treated as measurement error, which implies that the individual is assumed to follow the model exactly, or stated differently that the model represents the true state of the individual. This class of models can be stated as a state space model, which is written as

$$
\begin{aligned}
d\boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\phi})dt && (46)\\
\boldsymbol{y}_j &= \boldsymbol{h}(\boldsymbol{x}_j, \boldsymbol{u}_j, t_j, \boldsymbol{\phi}) + \boldsymbol{e}_j && (47)
\end{aligned}
$$

where $\boldsymbol{x}_t$ is the state (vector) in the model and the model for the state is given by (46). The parameters in the model are denoted $\boldsymbol{\phi}$, and $\boldsymbol{u}_t$ is a vector of input variables to the system. The second equation (47) in the state space model is the measurement equation defining how the states are observed. In this case an additive error model is chosen, but this is only one of several choices. Both the measurement and state equation can be multi-dimensional.

The states can represent amounts, concentrations, time-varying parameters or other dynamic parts of a system described by a state space model. At any point in time the state vector contains all the information about the future which is known as the Markovian property. The input variables $\boldsymbol{u}_t$ is a process influencing the system and it is observed only at measurement time points and is often assumed constant in between (known as zero order hold). A typical input variable could be e.g. body temperature or room temperature that may affect the system. The input process is assumed to be known exactly and as a consequence future values of the states in a deterministic model can be predicted without uncertainty by solving the ODE.

In the following a one-compartment ODE model for an intravenous bolus dose will be used as an example. The model is described by

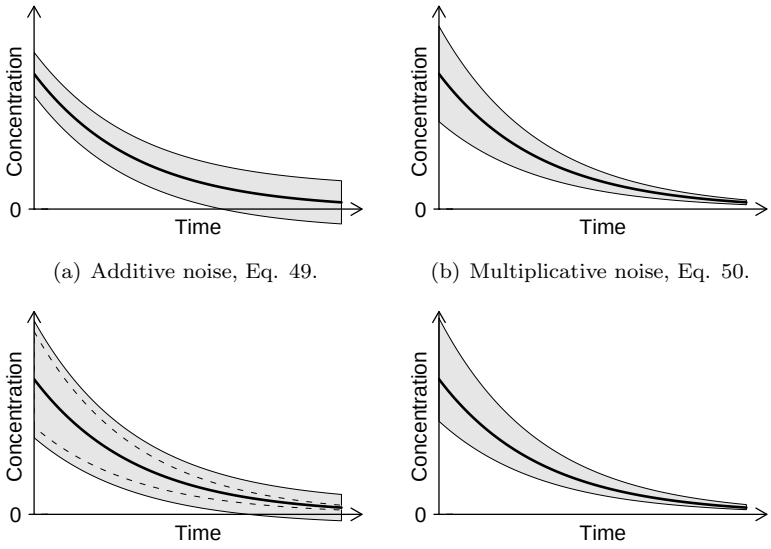
$$dx_t = -kx_t\ dt \tag{48}$$

which is also shown in (3). Here x_t represents concentration in the central compartment and k is the elimination rate constant.

Traditionally there are four main types of error models for the measurement equation called additive, multiplicative, additive and multiplicative, and log-normal error. The four error models are shown in (49), (50), (51) and (52) respectively, where e_j, $e_{j,1}$ and $e_{j,2}$ are Normal IID random variables. In Figure 13 the error models are illustrated graphically.

$$
\begin{aligned}
y_j &= x_j + e_j && (49)\\
y_j &= x_j \cdot (1 + e_j) && (50)\\
y_j &= x_j \cdot (1 + e_{j,1}) + e_{j,2} && (51)\\
y_j &= x_j \cdot \exp(e_j) && (52)
\end{aligned}
$$

The optimal choice of measurement error model is very dependent on the data. The additive error model (49) is the simplest but may not always be



(a) Additive noise, Eq. 49.

(b) Multiplicative noise, Eq. 50.

(c) Additive and multiplicative noise, Eq. 51.

(d) Log-normal noise, Eq. 52.

Figure 13: Error models illustrated with 95% prediction intervals around the model mean for an intravenous bolus dose.

appropriate. Measurements of concentrations are usually more uncertain for higher values, and this is not included in the additive model. However, if the measured concentrations only range over a small interval, the additive model may still be reasonable. When simulating from a model with concentrations close to zero the additive model will easily give negative values, see Figure 13(a), and this is often problematic.

The multiplicative error model (50) takes the increasing uncertainty for higher values into account. In this model the standard deviation of the residuals increases proportionally to the mean of the model. The model can still give negative values in simulations, but it is far less likely than with the additive model. In some cases it is appropriate to use a combination of the additive and multiplicative model, as specified in (51). The model still has increasing residual standard deviation with the mean but also allows for a larger uncertainty for smaller values.

The last error model shown in Figure 13 is the log-normal error model (52). This model resembles the multiplicative model as can be seen by comparing Figure 13(b) and 13(d). Being log-normal the residual distribution is asymmetric and bounded away from zero, which is an advantage for simulation as it will only give positive values. However, the distribution does not include 0, and

this is a problem if the observed data has such measurements. The log-normal error model can be achieved with an additive error structure by using a log-transformation of the observations, i.e. $\log y_j = \mu^* + e_j$ where $\mu^* = \log \mu$. An effect of this is that if the model for μ^* is additive then the resulting model for μ will be multiplicative since $\mu = \exp(\mu^*)$.

6.1.2 Error model using SDEs

In most cases it is not reasonable to assume that the variation in time of the concentration for an individual follows the model exactly as it is assumed using an ODE model. As mentioned earlier there may also be some variation due to incorrect model specification, true random biological variation or uncertainty from measuring an input process which cannot be explained or included in the ODE model. A way to describe such sources of errors is to base the state space model on stochastic differential equations (SDEs) instead of ODEs. This is called a stochastic state space model and is defined as

$$\begin{aligned} d\boldsymbol{x}_t &= \boldsymbol{f}(\boldsymbol{x}_t, \boldsymbol{u}_t, t, \boldsymbol{\phi}_i)dt + \boldsymbol{\sigma}_\omega(\boldsymbol{u}_t, t, \boldsymbol{\phi}_i)d\boldsymbol{\omega}_t & (53) \\ \boldsymbol{y}_j &= \boldsymbol{h}(\boldsymbol{x}_j, \boldsymbol{u}_j, t_j, \boldsymbol{\phi}_i) + \boldsymbol{e}_j \ . & (54) \end{aligned}$$

The stochastic differential equation in (53) is based on the Standard Wiener process ω_t (Øksendal 1992). This process is characterized by $\omega_0 = 0$, it is almost surely continuous and it has independent normal increments with $\omega_t - \omega_s \sim N(0, t-s)$ for $0 \leq s < t$. The Wiener process can be seen as a process that has the properties expected from the limit of a discrete random walk $\sum_{i=1}^{t/\Delta t} e_i$ with $e_i \sim N(0, \Delta t)$ for $\Delta t \to 0$.

Using the Wiener process the simple one-compartment elimination model in (48) can be extended to a model based on SDEs by writing

$$dx_t = -kx_t\ dt + \sigma_\omega d\omega_t \ . \quad (55)$$

The term $d\omega_t$ is an infinitesimal increment of the Wiener process and the model can thus in simple terms be described as an ordinary differential equation where the evolution in time is perturbed by normal distributed noise. The solution to (55) is

$$x_t = x_0 e^{-kt} + \int_0^t \sigma_\omega e^{-k(t-s)} d\omega_s \quad (56)$$

which can be seen to be the ODE solution plus the integral of Wiener process increments with exponential weights. The process is known as an Ornstein-Uhlenbeck process with zero mean. A simulation of the process is shown in Figure 14.

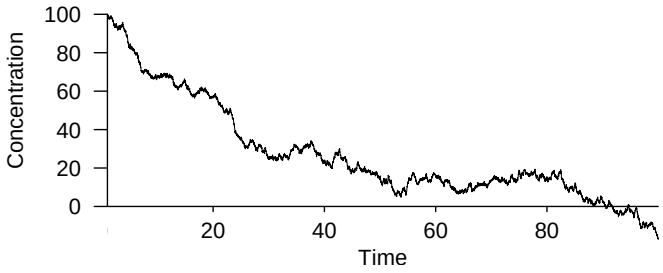


Figure 14: Simulation of simple one-compartment SDE model (55).

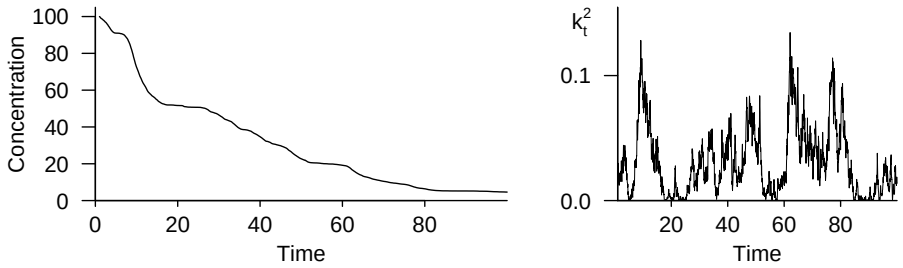
There are a few problems with the model proposed in (55). There is nothing limiting the concentration from increasing when the increments of the Wiener process are positive. This should normally not happen in the real world as this would require a reverse elimination process. Moreover the model will fluctuate around zero when the initial dose has been eliminated, and the model will thus predict negative concentrations. In fact it can be shown that the unconditional distribution of x_t is normal with mean zero which is not suitable in a model for concentrations.

A more realistic model can be achieved by adding noise to the elimination rate constant k instead of directly to the concentration. A good first choice is to model k_t as an Ornstein-Uhlenbeck process with a non-zero mean $\bar{k}$. However, the elimination rate should never be negative, so the elimination rate used will be k_t^2. The model is given by

$$dx_t = -k_t^2 x_t \, dt \tag{57}$$

$$dk_t = -\gamma(k_t - \bar{k})dt + \sqrt{2\sigma_\omega^2\gamma}d\omega_t \; . \tag{58}$$

It can be shown that the unconditional distribution of k_t is normal with mean $E[k_t] = \bar{k}$ and that k_t has the autocorrelation function $C_{k_t}(t) = \sigma_\omega^2 \exp(-\gamma t)$. Thus the elimination rate (k_t^2) will be χ^2-distributed with mean $E[k_t^2] = \bar{k}^2 + \sigma_\omega^2$. In Figure 15 a simulation of the model is shown. Since the elimination rate in Figure 15(b) is always non-negative, the concentration curve will be monotonely decreasing towards zero as would be expected in real life.



(a) Concentration as state 1.

(b) Elimination rate as state 2 squared.

Figure 15: Simulation of an extended one-compartment SDE model that fulfills some basic physiological constraints.

6.1.3 Case study: Advantages of using SDEs

In order to illustrate the advantages of modelling using SDEs a small simulation study will be performed.

It is assumed that the drug of interest is eliminated based on the model in (57) and (58). That is, the concentration is assumed to follow a first order elimination with an elimination rate that varies in a non-deterministic way. The optimal model is thus naturally the SDE model that generates the data, since the non-deterministic variations of the elimination rate cannot be modelled any further. However, the example will focus on illustrating the errors that are introduced if the data is modelled using the simple ODE model shown in (48). The residual error will be assumed to be independently log-normally distributed as defined in (52). Combining this the state space model used for generating data contains two states and one response variable and is thus given by

$$
\begin{aligned}
dx_t &= -k_t^2 x_t \, dt && (59)\\
dk_t &= -\gamma(k_t - \bar{k})dt + \sqrt{2\sigma_\omega^2 \gamma} d\omega_t && (60)\\
\log y_j &= \log x_j + e_j && (61)
\end{aligned}
$$

which will in short be denoted the SDE model. The parameters used for the simulation are $\bar{k} = 0.05$, $\gamma = 0.40$ and $\sigma_\omega = 0.15$, which gives an expected elimination rate of $E[k_t^2] = 0.025$ min^{-1} and an expected half-life of 28 minutes. The measurement variation is $e_j \sim N(0, S)$ where $S = 0.20^2$. Dose is 100mg and volume of distribution is 10L giving an initial concentration of 100mg/L. The individual is sampled with 5 minute intervals up to 100 minutes giving 21 observations. The simulated data is shown in Figure 16(a).

The data is modeled both using the original SDE model and an ODE model where the stochastic part is removed. The ODE model is thus a standard one-

compartment model with first order elimination. The ODE model is defined as

$$\begin{aligned} dx_t &= -kx_t\ dt \quad (62) \\ \log y_j &= \log x_j + e_j \quad (63) \end{aligned}$$

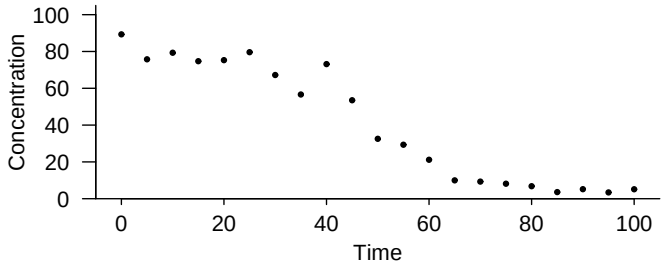
which is equivalent to the SDE model with $\sigma_\omega = 0$. Both the ODE model and SDE models are fitted using the maximum likelihood method, where the likelihood function is evaluated using the Kalman Filter. The estimation method is explained in more detail in Section 6.3. The estimate of the two states (concentration and elimination rate) in the SDE model can be found based on the estimated parameters using the so-called Kalman smoothing estimate. The results of the two model fits as given by the estimated concentration profiles are shown in Figure 16(b) and 16(c).

In order to judge the model fits the residuals for the ODE and SDE models are shown in Figure 16(d) and 16(e) and the auto-correlation functions for the residuals are shown in Figure 16(f) and 16(g).

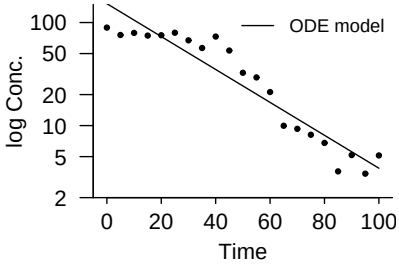
The first obvious observation of the difference between the fits is that the simple ODE model cannot capture the time varying elimination rate which gives rise to persistence in time of the residuals. This is not the case for the SDE model, which assumes a stochastic elimination rate and estimates it based on the model and data. The result is that the residuals are uncorrelated in time for SDE model whereas they are strongly auto-correlated for the ODE model. This in effect falsifies the ODE model in this case, as both models are based on an assumption of uncorrelated measurement error.

The problem is also apparent from the parameter estimates them selves. The estimates of the measurement variation in the two models are $\hat{S}_{ODE} = 0.39^2$ and $\hat{S}_{SDE} = 0.22^2$ which should be compared to the true value of $S = 0.20^2$. This shows that the time variation of the elimination rate has been included in the measurement error in the ODE model, since this is the only place it allows variation to enter, and hence a wrong interpretation of the measurement error is provided by the ODE model.

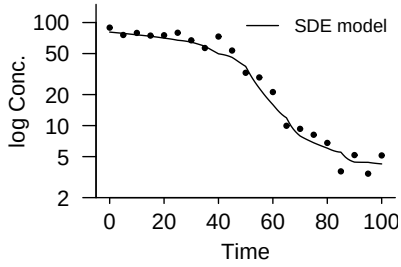
It is of interest to see if the estimate in the SDE model of the time varying behavior of the elimination rate (k_t^2) is accurate. Figure 17 shows the outcome of the elimination rate process from the simulation compared to the Kalman smoothing estimate from the SDE model and also the constant estimate from the ODE model. Generally it appears that the Kalman smoothing estimate is fairly close to the true elimination rate, but the accuracy will naturally always be dependent on the kind of measurement noise, sampling rate and appropriateness of the assumed model.



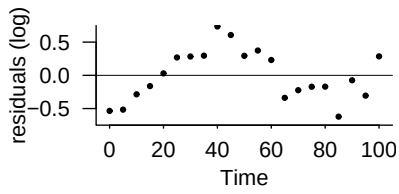
(a) Simulated data from model based on SDEs.



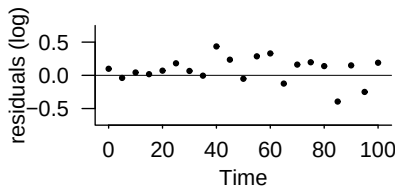
(b) Fitted ODE model on log-scale.



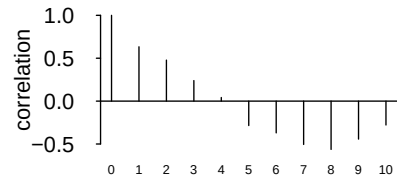
(c) Fitted SDE model on log-scale.



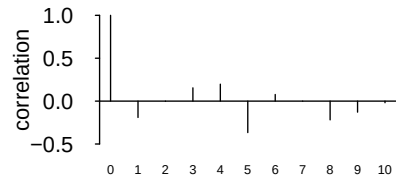
(d) Residuals in ODE model.



(e) Residuals in SDE model.



(f) ODE residual auto-correlation.



(g) SDE residual auto-correlation.

Figure 16: Comparison of fitted ODE and SDE models.

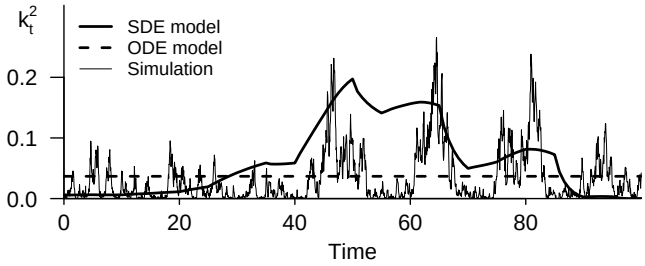


Figure 17: Comparison of estimated elimination rates.

6.1.4 Discussion of SDEs

The previous example in Section 6.1.3 has illustrated some of the issues that arise in a case where an ODE model provides an insufficient error structure. The estimate of the measurement variation was seen to be to high since it also included the stochastic variation of the system, and it resulted in auto-correlated residuals. Such situations calls for the use of an SDE model. In other situations an SDE model may be justified by a need to capture variation in the states introduced by actual stochastic behavior or a too simple deterministic part of a model or by variation caused by measurement error from the input process.

For estimation in the previous example both models were used in a maximum likelihood frame work. From a likelihood perspective, the failure of the ODE model occurs since it is not able to give a sufficient description of the 'true' likelihood function. Such an error cannot be ignored, as it in turn will invalidate e.g. likelihood ratio tests for model reduction and other classical statistical tests that may be used.

In many ways modeling using SDEs seems like an intuitive choice, since it facilitates a way to include dynamic biological variation in the model. Often however, the need for an SDE model is hidden by a sparse sampling scheme with few and distant observations, since it can be hard to detect residual autocorrelation in these situations. This may however change in the future with increasing use of modern frequent-sampling equipment within many areas, which very likely will reveal residual auto-correlation when using standard ODE models.

An example has been shown by Overgaard et al. (2007) using a PK/PD model of effects on thermo-regulation in monkeys. In this case an ODE model is shown to be insufficient, and only a model based on SDEs is able to provide a proper description of the error structure. Due to this the SDE model is able to give realistic simulations and predictions as opposed to the ODE model. Generally, in cases where the model should also be used for control purposes it is important that the model has realistic prediction properties. An example of an application for control of a biological system could be a model for controlling

the insulin secretion rate in diabetic patients.

Another application of SDEs in modelling biological systems is to use it as a tool for model development. In this context SDEs can be used as a tool to extract information from data about the appropriate model in situations with a complex underlying deterministic model structure. In Kristensen et al. (2005) a framework is presented for using an SDE model to allow tracking of time variations of parameters to reveal new functional relationships. Although this use aims at improving the deterministic part of a model, it may still result a final model based on SDEs if all stochastic components cannot replaced by deterministic relationships.

6.2 Multiple individuals

A typical data set for PK/PD modelling consists of data from several individuals who all have been exposed to a similar trial. The previous section has been concerned with modelling a single individual, but it is natural to include all individuals in the same model.

A mixed-effects model allows this by assuming the same model for each individual and by further assuming that the parameters in this model can vary between individuals. The general structure for data in a mixed-effects model is

$$\boldsymbol{y}_{ij}, \quad i = 1, ..., N, \quad j = 1, ..., n_i \tag{64}$$

which is an extension of the data structure in (45). The response $\boldsymbol{y}_{ij}$ is a vector of measurements at time t_{ij} for individual i, N is the number of individuals and n_i is the number of measurements for individual i.

In a mixed-effects model the variation is split into intra-individual variation and inter-individual variation, which is modelled by a first and second stage model. The first stage model is the (stochastic) state space model and the second stage model is given by

$$\boldsymbol{\phi}_i = g(\boldsymbol{\theta}, \boldsymbol{\eta}_i, \boldsymbol{Z}_i) \tag{65}$$

where $\boldsymbol{\phi}_i$ are the individual parameters for the first stage model. The random effects $\boldsymbol{\eta}_i$ have the distribution $\boldsymbol{\eta}_i \sim N(\boldsymbol{0}, \boldsymbol{\Omega})$, $\boldsymbol{\theta}$ are the fixed effects (also called population parameters) and $\boldsymbol{Z}_i$ are possible covariates. The second stage model will often look like $\phi = \theta + \eta$ or $\phi = \theta \cdot \exp \eta$ giving either a normal or log-normal distribution of parameters between individuals.

6.3 Estimation

Parameter estimation in the mixed effects model is most often facilitated by the maximum likelihood method. The population likelihood function is based on the distribution functions for the first and second stage models denoted p_1 and p_2, respectively. The distribution function for the second stage model is simply the normal distribution. For the first stage model based on the stochastic state space model the likelihood function can be evaluated by using the Kalman filter (Overgaard et al. 2005). The population likelihood function for the fixed effects are found by integration over the random effects. This is given as

$$L(\boldsymbol{\theta}) \quad = \quad \prod_{i=1}^{N} \int p_1(\mathcal{Y}_{in_i}|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) d\boldsymbol{\eta}_i = \prod_{i=1}^{N} \int \exp(l_i) d\boldsymbol{\eta}_i \tag{66}$$

where $\mathcal{Y}_{in_i}$ are all observations for individual i and l_i is the *a posteriori* log-likelihood function for the ith individual. The population likelihood function in (66) can not be evaluated analytically, and therefore l_i is approximated by a second-order Taylor expansion, where the expansion is made around the value $\boldsymbol{\eta}_i^*$ that maximizes $l_i(\boldsymbol{\eta}_i)$ in the value l_i^*. At this optimum the first derivative $\boldsymbol{\nabla} l_i\big|_{\boldsymbol{\eta}_i^*} = 0$ and the population likelihood function therefore reduces to

$$L(\boldsymbol{\theta}) \quad \approx \quad \prod_{i=1}^{N} \left| \frac{-\boldsymbol{\Delta} l_i^*}{2\pi} \right|^{-\frac{1}{2}} \exp(l_i^*) \tag{67}$$

as shown in Appendix A. The approximation of the 2nd derivative at the optimum $\boldsymbol{\Delta} l_i^*$ is obtained using the First-Order Conditional Estimation (FOCE) method, which is defined as

$$\boldsymbol{\Delta} l_i^* \approx -\sum_{j=1}^{n_i} \left(\boldsymbol{\nabla} \boldsymbol{\epsilon}_{ij}^T R_{i(j|j-1)}^{-1} \boldsymbol{\nabla} \boldsymbol{\epsilon}_{ij} \right) - \boldsymbol{\Omega}^{-1} \quad , \quad \boldsymbol{\nabla} \boldsymbol{\epsilon}_{ij} = \frac{\partial}{\partial \boldsymbol{\eta}_i} \boldsymbol{\epsilon}_{ij} \Big|_{\boldsymbol{\eta}_i^*} .$$

In cases where the fist stage model is non-linear as in the state space model in (53) and (54) the combined model is called a non-linear mixed effects model. When working with SDEs there are only a few software tools available that are able to estimate parameters in this class of models. These tools are listed below.

6.3.1 NONMEM

NONMEM is a software package developed at University of California, San Francisco (UCSF) for use in population PK/PD modelling (Beal and Sheiner

2004). It first appeared in 1979 and its name is an acronym for non-linear mixed effects modeling. NONMEM has become the defacto standard software tool used for PK/PD modelling as it is a very flexible tool and well tested throughout many years of development. NONMEM is however only intended for modelling based on ODEs but it is possible to make it estimate models based on SDEs as shown by Tornøe et al. (2005). This is basically done by including the Kalman filter into the model definition, but the Kalman filter has to be derived and implemented for every new model that is created, and it is thus cumbersome to work with and only feasible for simple models.

6.3.2 CTSM

CTSM is a program for performing estimation of state space models based on SDEs (Kristensen and Madsen 2003, Kristensen et al. 2004). The program is intended for single subject modelling but also handles multiple subjects based on a pooled likelihood without random effects. It has been developed at DTU Informatics. CTSM has previously been used for PK/PD modelling using SDEs in e.g. Tornøe et al. (2004a, 2004b) and Kristensen et al. (2005).

6.3.3 PSM

PSM in an acronym for Population Stochastic Modelling and is a software package developed at DTU Informatics (Mortensen et al. 2007, Klim et al. 2008). It is like NONMEM aimed at non-linear mixed effects modelling but it is focused on modelling using SDEs and as opposed to NONMEM it also directly handles a multivariate response. PSM supports a typical PK data structure with dosing information, co-variates and also missing observations. PSM is freely available as an extension package for R, which is a free software environment for statistical computing. Instructions for download and installation in R can found at `http://www.imm.dtu.dk/psm`.

Appendix

A NLME log-likelihood function

The Non-linear mixed effects likelihood function is defined as

$$L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) = \prod_{i=1}^{N} \int p_1(\mathcal{Y}_{in_i}|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega}) d\boldsymbol{\eta}_i \tag{68}$$

$$= \prod_{i=1}^{N} \int L_i(\boldsymbol{\eta}_i) d\boldsymbol{\eta}_i \tag{69}$$

where L_i is the individual *a posteriori* likelihood function. In most cases the integral cannot be evaluated analytically. For a general evaluation the individual *a posteriori* likelihood function can be approximated by a second order Taylor series expansion of $\log(L_i)$ around the value $\boldsymbol{\eta}_i^*$ which maximizes $\log(L_i(\boldsymbol{\eta}_i))$. Also $l_i = \log(L_i)$, $L_i^* = \exp(l_i^*) = L_i(\boldsymbol{\eta}_i^*)$, $\boldsymbol{\nabla} l_i^* = \frac{\partial}{\partial \boldsymbol{\eta}_i} l_i \Big|_{\boldsymbol{\eta}_i^*}$, $\boldsymbol{\Delta} l_i^* = \frac{\partial^2}{\partial \boldsymbol{\eta}_i \partial \boldsymbol{\eta}_i^T} l_i \Big|_{\boldsymbol{\eta}_i^*}$. It follows that

$$l_i(\boldsymbol{\eta}_i) \approx l_i^* + \boldsymbol{\nabla} l_i^{*T}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*) + \frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^T \boldsymbol{\Delta} l_i^* (\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*) \tag{70}$$

$$\approx l_i^* + \frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^T \boldsymbol{\Delta} l_i^* (\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*) \tag{71}$$

$$L_i(\boldsymbol{\eta}_i) \approx L_i^* \exp\left(-\frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^T (-\boldsymbol{\Delta} l_i^*)(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)\right) \tag{72}$$

since $\boldsymbol{\nabla} l_i = 0$ at $\boldsymbol{\eta}_i^*$. Based on the approximation the integral can now be evaluated by moving constants such that the integral is over a Gaussian density with mean $\boldsymbol{\eta}_i^*$ and co-variance $(-\boldsymbol{\Delta} l_i^*)^{-1}$. The result is

$$\int L_i(\boldsymbol{\eta}_i) d\boldsymbol{\eta}_i \approx \int L_i^* \exp\left(-\frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^T (-\boldsymbol{\Delta} l_i^*)(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)\right) d\boldsymbol{\eta}_i \tag{73}$$

$$\approx L_i^* \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i^*}\right|^{\frac{1}{2}} \int \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i^*}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)^T (-\boldsymbol{\Delta} l_i^*)(\boldsymbol{\eta}_i - \boldsymbol{\eta}_i^*)\right) d\boldsymbol{\eta}_i$$

$$\approx L_i^* \left|\frac{2\pi}{-\boldsymbol{\Delta} l_i^*}\right|^{\frac{1}{2}} \tag{74}$$

$$\approx L_i^* \left|\frac{-\boldsymbol{\Delta} l_i^*}{2\pi}\right|^{-\frac{1}{2}} \tag{75}$$

where the step in Eq. (75) is taken to avoid a matrix inversion of the Hessian. The NLME log-likelihood function can now be approximated by

$$L(\boldsymbol{\theta}|\mathcal{Y}_{Nn_i}) \approx \prod_{i=1}^{N} L_i^* \left|\frac{-\boldsymbol{\Delta} l_i^*}{2\pi}\right|^{-\frac{1}{2}} . \tag{76}$$

References

Beal, S. L. and L. B. Sheiner (2004). *NONMEM® Users Guide*. NONMEM Project Group, University of California, San Francisco.

Gabrielsson, J. and D. Weiner (1997). *Pharmacokinetic and Pharmacodynamic Data Analysis: Concepts and Applications* (Second ed.). Kristianstads Boktryckeri.

Gibb, I. A. and B. J. Anderson (2008). Paracetamol (acetaminophen) pharmacodynamics: interpreting the plasma concentration. *Arch. Dis. Child. 93*(3), pp. 241–247.

Klim, S., S. B. Mortensen, N. R. Kristensen, R. V. Overgaard, and H. Madsen (2008). Population stochastic modelling (PSM) - An R package for mixed-effects models based on stochastic differential equations. Technical report, DTU Informatics. Submitted to *Comput. Methods Programs Biomed.*

Kristensen, N. R. and H. Madsen (2003). Continuous time stochastic modelling - CTSM 2.3 Mathematics guide. Technical report, Technical University of Denmark.

Kristensen, N. R., H. Madsen, and S. H. Ingwersen (2005). Using stochastic differential equations for pk/pd model development. *J. Pharmacokinet. Pharmacodyn.. 32*, pp. 109–41.

Kristensen, N. R., H. Madsen, and S. B. Jørgensen (2004). Parameter estimation in stochastic grey-box models. *Automatica 40*, pp. 225–237.

Mortensen, S. B., S. Klim, B. Dammann, N. R. Kristensen, H. Madsen, and R. V. Overgaard (2007). A matlab framework for estimation of nlme models using stochastic differential equations: applications for estimation of insulin secretion rates. *J. Pharmacokinet. Pharmacodyn. 34*(5), pp. 623–642.

Overgaard, R. V., N. Holford, K. A. Rytved, and H. Madsen (2007). PKPD model of interleukin-21 effects on thermoregulation in monkeys–application and evaluation of stochastic differential equations. *Pharm. Res. 24*(2), pp. 298–309.

Overgaard, R. V., N. Jonsson, C. W. Tornøe, and H. Madsen. (2005). Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm. *J. Pharmacokinet. Pharmacodyn. 32*(1), pp. 85–107.

Rawlins, M. D., D. B. Henderson, and A. R. Hijab (1977). Pharmacokinetics of paracetamol (acetaminophen) after intravenous and oral administration. *Eur. J. Clin. Pharmacol. 11*(4), pp. 283–286.

Rowland, M. and T. N. Tozer (1997). *Clinical Pharmacokinetics - Concepts and Applications* (Third ed.). Lippincott Williams & Wilkins.

Tornøe, C. W., J. L. Jacobsen, and H. Madsen (2004a). Grey-box pharmacokinetic/pharmacodynamic modelling of a euglycaemic clamp study. *J. Math. Biol. 48*(6), pp. 591–604.

Tornøe, C. W., J. L. Jacobsen, O. Pedersen, T. Hansen, and H. Madsen (2004b). Grey-box modelling of pharmacokinetic/pharmacodynamic systems. *J. Pharmacokinet. Pharmacodyn. 31*(5), pp. 401–417.

Tornøe, C. W., R. V. Overgaard, H. Agersø, H. A. Nielsen, H. Madsen, and E. N. Johnson (2005). Stochastic differential equations in NONMEM®: Implementation, application, and comparison with ordinary differential equations. *Pharm. Res. 22*(8), pp. 1247–1258.

Øksendal, B. (1992). *Stochastic differential equations (3rd ed.): an introduction with applications*. New York, NY, USA: Springer-Verlag New York, Inc.

Appendix C

Paper C

Title:
Population stochastic modelling (PSM) - An R package for mixed-effects models based on stochastic differential equations.

Authors:
S. Klim, S. B. Mortensen, N. R. Kristensen, R. V. Overgaard, and H. Madsen.

Published in:
Computer Methods and Programs in Biomedicine 94, pp. 279-289 (2009).

COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 94 (2009) 279–289

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

Population stochastic modelling (PSM)—An R package for mixed-effects models based on stochastic differential equations

Søren Klim[a,c,*], *Stig Bousgaard Mortensen*[b,c], *Niels Rode Kristensen*[a], *Rune Viig Overgaard*[a], *Henrik Madsen*[c]

[a] *Novo Nordisk A/S, Novo Alle, 2880 Bagsværd, Denmark*
[b] *H. Lundbeck A/S, Ottiliavej 9, 2500 Valby, Denmark*
[c] *Department of Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark*

ARTICLE INFO

Article history:
Received 26 August 2008
Received in revised form
30 January 2009
Accepted 1 February 2009

Keywords:
Stochastic differential equations (SDEs)
State-space models
Mixed-effect
Pharmacokinetic
Pharmacodynamic

ABSTRACT

The extension from ordinary to stochastic differential equations (SDEs) in pharmacokinetic and pharmacodynamic (PK/PD) modelling is an emerging field and has been motivated in a number of articles [N.R. Kristensen, H. Madsen, S.H. Ingwersen, Using stochastic differential equations for PK/PD model development, J. Pharmacokinet. Pharmacodyn. 32 (February(1)) (2005) 109–141; C.W. Tornøe, R.V. Overgaard, H. Agersø, H.A. Nielsen, H. Madsen, E.N. Jonsson, Stochastic differential equations in NONMEM: implementation, application, and comparison with ordinary differential equations, Pharm. Res. 22 (August(8)) (2005) 1247–1258; R.V. Overgaard, N. Jonsson, C.W. Tornøe, H. Madsen, Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm, J. Pharmacokinet. Pharmacodyn. 32 (February(1)) (2005) 85–107; U. Picchini, S. Ditlevsen, A. De Gaetano, Maximum likelihood estimation of a time-inhomogeneous stochastic differential model of glucose dynamics, Math. Med. Biol. 25 (June(2)) (2008) 141–155].

PK/PD models are traditionally based ordinary differential equations (ODEs) with an observation link that incorporates noise. This state-space formulation only allows for observation noise and not for system noise. Extending to SDEs allows for a Wiener noise component in the system equations. This additional noise component enables handling of autocorrelated residuals originating from natural variation or systematic model error. Autocorrelated residuals are often partly ignored in PK/PD modelling although violating the hypothesis for many standard statistical tests.

This article presents a package for the statistical program R that is able to handle SDEs in a mixed-effects setting. The estimation method implemented is the FOCE[1] approximation to the population likelihood which is generated from the individual likelihoods that are approximated using the Extended Kalman Filter's one-step predictions.

* *Corresponding author at*: DTU Informatics, Technical University of Denmark, Richard Petersens Plads, Building 321, 2800 Kgs. Lyngby, Denmark. Tel.: +45 4525 3351.
E-mail addresses: SKli@novonordisk.com, skl@imm.dtu.dk (S. Klim).

[1] FOCE—First-Order Conditional Estimation.

0169-2607/$ – see front matter © 2009 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2009.02.001

1. Introduction

The use of mixed-effects models based on ordinary differential equations (ODEs) is the standard for pharmacokinetic and pharmacodynamic (PK/PD) modelling. The use of stochastic differential equations (SDEs) is an emerging field and has been introduced and motivated in the papers [1–4]. This paper presents an accessible software package for handling dynamic models based on SDEs in a mixed-effects setting. The program is a package for the statistical program R and thereby easy to install through R's interface and available for a wide range of operating systems.

The package implements the (Extended) Kalman Filter for evaluating the likelihood function in models based on SDEs. The parameter estimation procedure in the package is maximum likelihood based with fixed effects estimation based on the FOCE approximation.

2. Computational methods and theory

The most widely used program to analyze state-space models in PK/PD-modelling is NONMEM [5], which is focused on mixed-effects models based on ordinary differential equations. The use of SDEs in non-linear mixed-effects models is possible in NONMEM as described in [2]. The trick is an implementation of the Extended Kalman Filter in the NONMEM control file with corresponding adjustments to the data file. This is a non-trivial task even for rather simple models and must be repeated for every change in model or data. Single subject data can be modelled with stochastic differential equations in the program *CTSM* [6]. *CTSM* is a stand alone program that works across different platforms.

The matlab framework described in [7] handles SDEs in a mixed-effect setting. The experiences collected in the development of the Matlab framework have now been used to create an extended and more flexible R-package PSM.

The mathematical basis for the PSM package is also described in the articles [3,7,8]. It should be noted that there are notation differences between the articles.

For simplicity this article focuses on the class of linear models but it must be emphasised that the package also handles non-linear models.

2.1. Single subject

The modelling of observations for a single subject is based on a continuous-discrete state-space model. The states represent the internal hidden states of the system. The states reside in a continuous time domain and their dynamics are described by stochastic differential equations. The observations are sampled at discrete time points and hence described by a discrete time relation.

The class of linear models handled by PSM are time-invariant models meaning that system matrices do not change over time. More specifically the linear state-space model can be stated as

$$d\mathbf{x}_t = (\mathbf{A}(\boldsymbol{\phi}_i)\mathbf{x}_t + \mathbf{B}(\boldsymbol{\phi}_i)\mathbf{u}_t)\,dt + \boldsymbol{\sigma}(\boldsymbol{\phi}_i)d\boldsymbol{\omega}_t \quad (1)$$

$$\mathbf{y}_{ik} = \mathbf{C}(\boldsymbol{\phi}_i)\mathbf{x}_{ik} + \mathbf{D}(\boldsymbol{\phi}_i)\mathbf{u}_{ik} + \boldsymbol{e}_{ik} \quad (2)$$

where $\mathbf{x}_t \in \mathbb{R}^{dimX}$ is the vector of states at time t. The dimension of $\mathbf{x}$ is denoted as $dimX$ for simplicity. $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are time-invariant matrices defined as functions of $\boldsymbol{\phi}_i$ with properties $\mathbf{A}(\cdot) \in \mathbb{R}^{dimX \times dimX}$, $\mathbf{B}(\cdot) \in \mathbb{R}^{dimX \times dimU}$, $\mathbf{C}(\cdot) \in \mathbb{R}^{dimY \times dimX}$ and $\mathbf{D}(\cdot) \in \mathbb{R}^{dimY \times dimU}$. $\boldsymbol{\phi}_i$ is the parameter vector for the ith individual (see Eq. (9) for further details). The exogenous input $\boldsymbol{u} \in \mathbb{R}^{dimU}$ can be used to include other measured variables which influence the time evolution of the states in the model. The input $\boldsymbol{u}$ is assumed to be constant between observation points which is often refered to as zero-order hold or piece wise constant. The component $\boldsymbol{\sigma}(\boldsymbol{\phi}_i)d\boldsymbol{\omega}_t$ is the system noise consisting of a scaling diffusion term $\boldsymbol{\sigma}(\cdot) \in \mathbb{R}^{dimX \times dimX}$ and $\boldsymbol{\omega}_t$ which is a $dimX$-dimensional Wiener process. The subscript i denotes the ith subject and the subscript k is a short hand notation for t_k. $\mathbf{y}_{ik}$ is the observation at time t_k for the ith subject. $\boldsymbol{e}_{ik}$ is the residual for individual i at time t_k and is assumed to normal distributed $N(\mathbf{0}, \mathbf{S}(\phi_i))$ with $\mathbf{S}(\cdot) \in \mathbb{R}^{dimY \times dimY}$ being the covariance matrix for the errors.

2.2. Kalman Filter

The deterministic behaviour of ordinary differential equations can be handled with standard differential equation solvers. The additional component in the SDE systems requires a more advanced solution method. As mentioned in the introduction this package uses the Kalman Filter as solution method for systems of SDEs.

The Kalman Filter is only briefly explained in this article but the mathematics behind the Kalman Filter is well described in the Mathematics guide to CTSM [6] and in the original reference [9]. Several links and additional material can be found on the homepage [10].

The assumptions on system noise being driven by a Wiener process and normally distributed errors will in a linear system under some regularity conditions [6] result in the conditional densities for the observations being fully described by their first- and second-order moments. The Kalman Filter can be used to determine the optimal internal states in the system conditioning on prior observations. The Kalman Filter updates the internal state vector after each observation and during this process the Kalman Filter needs to weigh the probability of the residual being due to system noise or measurement noise. For this purpose the one-step prediction $\hat{\mathbf{y}}_{k|k-1}$ and associated covariance $\mathbf{R}_{k|k-1}$ are defined below:

$$\hat{\mathbf{y}}_{k|k-1} = E[\mathbf{y}_k|\mathcal{Y}_{k-1}, \boldsymbol{\phi}_i] \quad (3)$$

$$\mathbf{R}_{k|k-1} = V[\mathbf{y}_k|\mathcal{Y}_{k-1}, \boldsymbol{\phi}_i] \quad (4)$$

where $\mathcal{Y}_{k-1}$ denotes all measurements up to time t_{k-1}.

The description of conditional densities based on first- and second-order moments is only exact for linear models. For nonlinear models the Extended Kalman Filter can be used which is based on continous linerazations of the model however the forming of the conditional densitites will only be approximate.

The structure of the Kalman Filter is thus an iterative process with a prediction/updating scheme. The one-step

Table 1 – The Kalman Filter written in algorithmic form. Copied from [11].

Algorithm: The Kalman Filter
Given parameters and initial prediction
ϕ_i, $\hat{x}_{1|0}$ and $P_{1|0}$
For $k = 1$ to n_i **do**
Output Prediction:
$\hat{y}_{k|k-1} = C\hat{x}_{k|k-1} + Du_k$
$R_{k|k-1} = CP_{k|k-1}C^T + S$
State Update:
$K_k = P_{k|k-1}C^T R^{-1}_{k|k-1}$
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(y_k - \hat{y}_{k|k-1})$
$P_{k|k} = P_{k|k-1} - K_k R_{k|k-1} K_k^T$
State Prediction:
$d\hat{x}_{t|k}/dt = A\hat{x}_{t|k} + Bu_t$
$dP_{t|k}/dt = A_t P_{t|k} + P_{t|k}A^T + \sigma\sigma^T$
end for

prediction will be based on the deterministic part of the model as the Wiener component has mean zero. The measured observation should be considered as a result of the current states including accumulated system noise since last observation and current observation noise. An updating of the internal states based on the residual is weigthed according to system noise and measurement noise. The iterative structure restarts with a prediction based on the updated states.

The initial conditions for the Kalman Filter are the initial states $\hat{\mathbf{x}}_{1|0}$ and initial state covariance matrix $\hat{\mathbf{P}}_{1|0}$. Optimally the initial conditions and uncertainties are known a priori but generally none or only the initial state is known. The initial states can be either specified directly or estimated simultaneously with the remaining parameters. As the uncertainty is rarely known the package assumes that the initial state uncertainty is equal to the integral of the Wiener noise over a time interval equal to the interval between the first two observations.

$$\mathbf{P}_0 = P_s \int_{t_0}^{t_1} e^{As}\boldsymbol{\sigma}\boldsymbol{\sigma}^T \left(e^{As}\right)^T \mathrm{d}s \tag{5}$$

where P_s is a scaling factor. One solution method for the integral (5) is shown in [6].

In Table 1 the iterative structure of the Kalman Filter is shown in algorithmic form.

The Kalman Filter setup requires a specific model structure as shown in Eqs. (1) and (2). Two requirements that should be noted are the additive noise in the observations and the state independent Wiener component. The Kalman Filter cannot handle a multiplicative or full error model in Eq. (2). Using a log transformation of the observations a log normal error model can however be dealt with. The limitation with state independent Wiener component can only be surpassed by transformation of the system equations or by introducing more sophisticated methods such as higher order filters or Markov Chain Monte Carlo methods.

The residual used in the likelihood function is defined as

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \hat{\mathbf{y}}_{k|k-1} \tag{6}$$

The likelihood of the parameters $\boldsymbol{\phi}_i$ based on data can be calculated using the assumption of normality combined with the conditional covariance which is calculated in the Kalman Filter.

$$L(\boldsymbol{\phi}_i; \mathcal{Y}) = \left(\prod_{k=1}^{n_i} \frac{\exp(-(1/2)\boldsymbol{\epsilon}_k^T \mathbf{R}^{-1}_{k|k-1}\boldsymbol{\epsilon}_k)}{\sqrt{\det(\mathbf{R}_{k|k-1})}(\sqrt{2\pi})^{dimY}} \right) p(\mathbf{y}_0|\boldsymbol{\phi}_i) \tag{7}$$

The negative log likelihood can be derived from Eq. (7) by conditioning on the initial condition $\mathbf{y}_0$. The negative log-likelihood is the objective function used in the parameter estimation in the Kalman Filter.

$$-\ln(L(\boldsymbol{\phi}_i; \mathcal{Y}|\mathbf{y}_0)) = \frac{1}{2}\sum_{k=1}^{n_i} (\ln(\det(\mathbf{R}_{k|k-1})) + \boldsymbol{\epsilon}_k^T \mathbf{R}^{-1}_{k|k-1}\boldsymbol{\epsilon}_k) + \frac{1}{2}\left(\sum_{k=1}^{n_i} dimY\right) \ln(2\pi) \tag{8}$$

2.3. *Mixed-effects*

The use of non-linear mixed effects models in PK/PD modelling has long been the standard and has been supported by the functionality in NONMEM. The mixed-effects approach to model variation in pharmacokinetics was first introduced by Sheiner in [12]. Mixed-effects modelling is a hierarchical division of the variation, where the fixed effects describe the population mean and the random effects describe the inter-individual variation. This is often described in two stages. The first stage model describes the intra-individual variability and the second stage describes the inter-individual variation.

The first stage model is described in Eqs. (1) and (2) which are based on the individual parameters. The inter-individual variation in parameters is covered in the creation of the individual parameters in the function $h(\cdot)$ described below:

$$\boldsymbol{\phi}_i = h(\boldsymbol{\theta}, \boldsymbol{\eta}_i, \mathbf{Z}_i) \tag{9}$$

where $\boldsymbol{\theta}$ are the fixed effects—also sometimes referred to as the population parameters. $\mathbf{Z}_i$ denotes subject covariates and $\boldsymbol{\eta}_i \in N(\mathbf{0}, \boldsymbol{\Omega})$ are the random effects. The individual parameters can be modelled as either normally or log-normally distributed by combining the population parameters and the random effects in either an additive ($\phi_i = \theta + \eta_i$) or an exponential transform ($\phi_i = \theta \exp(\eta_i)$).

The likelihood function for the fixed effects can be stated as below:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N} \int p_1(\mathcal{Y}_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i) p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega})\, \mathrm{d}\boldsymbol{\eta}_i = \prod_{i=1}^{N} \int \exp(l_i)\, \mathrm{d}\boldsymbol{\eta}_i \tag{10}$$

where N is the number of subjects. $p_1(\mathcal{Y}_i|\boldsymbol{\theta}, \boldsymbol{\eta}_i)$ is the probability for the first stage model which is proportional to Eq. (7). $p_2(\boldsymbol{\eta}_i|\boldsymbol{\Omega})$ is the probability related to the second stage model that relates the random effects to the inter-individual variation. l_i is the *a posteriori* log-likelihood function for the ith individual. $\mathcal{Y}_i$ is the complete sequence of observations for individual i. The population likelihood function in Eq. (10) rarely has a closed form solution and therefore l_i is approximated by a second-order

Taylor expansion, where the expansion is made around the value $\hat{\boldsymbol{\eta}}_i$ that maximizes l_i. At this optimum the first derivative $\nabla l_i|_{\hat{\eta}_i} = 0$ and the population likelihood function will under some assumptions reduce to

$$L(\boldsymbol{\theta}) \approx \prod_{i=1}^{N} \left| \frac{-\boldsymbol{\Delta} l_i}{2\pi} \right|^{-(1/2)} \exp(l_i)|_{\hat{\eta}_i} \tag{11}$$

The approximation of the 2nd derivative $\boldsymbol{\Delta} l_i$ is done using the First-Order Conditional Estimation (FOCE) method, which is defined as

$$\boldsymbol{\Delta} l_i^* \approx -\sum_{j=1}^{n_i} (\nabla \boldsymbol{\epsilon}_{ij}^{\mathrm{T}} \mathbf{R}_{i(j|j-1)}^{-1} \nabla \boldsymbol{\epsilon}_{ij}) - \boldsymbol{\Omega}^{-1}, \quad \text{where} \quad \nabla \boldsymbol{\epsilon}_{ij} = \frac{\partial}{\partial \boldsymbol{\eta}_i} \boldsymbol{\epsilon}_{ij}|_{\boldsymbol{\eta}_i^*} \tag{12}$$

When the random effects have a non-linear influence on the likelihood through the first stage model, the combined model is called a non-linear mixed effects model.

The conditional residual covariance $\mathbf{R}_{i(j|j-1)}^{-1}$ is calculated in the (Extended) Kalman Filter and the gradient in the residual with relation to the random effects $\nabla \epsilon_{ij}$ is typically found by numerical methods.

2.4. *Maximum likelihood estimation*

The population likelihood function in (11) is used in maximum likelihood estimation of the population parameters. This optimization becomes a nested optimization as the FOCE approximation is based on the optimal random effects ($\boldsymbol{\eta}_i^*$). Each calculation of the population likelihood thus requires N optimizations of the random effects. This nested optimization makes the computational effort substantial and highly dependent on the number of subjects, number of observations and the number of fixed and random effects. The optimization in the `PSM` package is performed with the default optimizer (`optim`) which is a quasi-Newton based optimizer.

The optimization can be constrained using boundaries on the population parameters using a logit-transform. It is assumed that the optimizer works on a unconstrained parameter space so the logit transform maps the bounded parameters into an unbounded space. In order to avoid flat gradients in population parameters in the outer parts of the logit transform a penalty function has been added. The penalty function is defined as below:

$$P(\lambda, \theta, \theta^{\min}, \theta^{\max}) = \lambda \left(\sum_{j=1}^{p} \frac{|\theta_j^{\min}|}{\theta_j - \theta_j^{\min}} + \sum_{j=1}^{p} \frac{|\theta_j^{\max}|}{\theta_j^{\max} - \theta_j} \right) \tag{13}$$

where p is the number of parameters and $\theta_j^{\min}$ and $\theta_j^{\max}$ denotes the lower and upper limit for the jth parameter.

The computational effort in the parameter estimation can as already mentioned be substantial and it is advised to find good initial estimates for the parameters in advance.

3. Program description

The framework for handling mixed-effects models based on SDEs has previously been implemented in Matlab [7]. The `R` package `PSM` presented here is a ported and extended version. The switch in platform was motivated by `R` being an open source program and its availability on different platforms. The `PSM` package was extended from the Matlab framework by extending the flexibility, improving performance by low level implementations and enabling capabilities for bolus doses. The dosing capability is crucial for modelling in drug development.

The package is able to handle multivariate observations, which are useful when performing simultaneous fits of multivariate data such as insulin and glucose. Another feature is that it is possible to have input into the model and include subject covariates. Finally, the package handles missing observations.

The package is mainly implemented in the `R`-language which is closely related to the `S`-language. Core components of the code have been implemented in `FORTRAN` for faster computation.

The current `PSM` version is 0.8-3. The package has dependencies for three other `R` packages. `MASS` is used in the simulation part to sample from the multivariate normal distribution. `MASS` is an integrated part of the `R` installation. The package `odesolve` is used in the non-linear models to solve systems of differential equations. The package `numDeriv` is used to calculate the Hessian which is used in the calculations of the confidence intervals for the estimated parameters.

The `PSM` package is available as a standard `R` package. Installation can be performed using the `R` interface or by executing the command.

```
> install.packages("PSM")
```

The package comes with complete documentation and "get-started" guide. The documentation can be found by executing the command `help("PSM")`. A more thorough guide to the package and its usage can be obtained with the command

```
> vignette("PSM")
```

The package is divided into three parts according to functionality. The three parts are

- Simulation
- Estimation
- Smoothing

Simulation is the creation of observations based on a given model and model inputs. The Estimation part performs a maximum likelihood estimation of the population parameters based on the one-step predictions in the Kalman Filter. The smoothing functionality creates the optimal state estimates based on the entire data series and a set of parameters.

All three functions operate on a model object and a data object. The following sections introduce the model and data objects before going into detail with the three functions.

Table 2 – Model specification.

Functions	Output
`Matrices`	List with system matrices see Eq. (1)
`X0`	Matrix with initial state condition(s)
`SIG`	Matrix with diffusion scaling term σ
`S`	Matrix with residual covariance
`h`	Vector with individualized parameters ϕ_i
`ModelPar`	List with fixed effects θ and inter individual variation Ω

Table 3 – Data specification.

PSMnotation	Description
`Time`	Vector with dose and observation times
`Y`	Matrix with observations. Multivariate observations in columns
`U`	Matrix with input
`Dose`	List with `Time`, `State`and `Amount`
`covar`	Subject covariates

3.1. *Model specification*

The model specification is divided into components corresponding to the mathematical parts of the state-space model. For the linear case the state-space model can be stated in matrix form as seen in Eq. (1), but variance components and initial conditions are also needed. Table 2 shows the components in the model. The components are collected in a list to have a single object that contains the model. The individual components are all functions that return either a matrix or a list, if multiple outputs are needed.

Fig. 1 shows the model specification in a diagram with mathematical references displayed. The sequence for these components needs some elaboration. The `ModelPar` function is used to split the vector of parameters to be optimized Θ into the fixed effects vector θ and the random effects covariance matrix Ω. The individual parameters are created with the h function that uses the fixed effects, the random effects and the subject covariates to create the ϕ_i vector. The remaining components in the dynamical system can be evaluated using ϕ_i and the system input u.

3.2. *Data specification*

The data specification in the simulation procedure is different from the specification in the parameter estimation and smoothing. In the simulation part the observations are simulated based on the model. Time points for the observations and potentially system input, doses or subject covariates still need to be provided. The time points, system input, doses or covariates are specified per subject in a list. Names in the list need to be according to the `PSM` specifications as the referencing in the package is done with names. The naming of the components can be seen in Table 3. The lists for all subjects are finally collected in a list which makes the overall data object a list of lists.

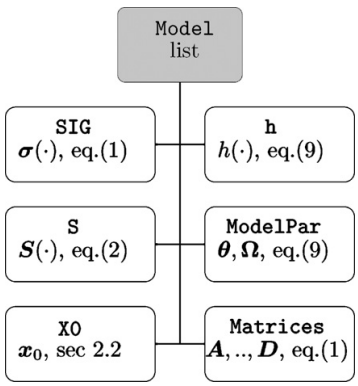


Fig. 1 – Model components.

The observations are specified in the element `Y` which is a matrix with dimensions [$dimY$, $dimT$]. $dimT$ is the length of the `Time` vector. As can be seen from the dimensions of $\mathbf{Y}$, multivariate observations are specified in columns. Missing observations are indicated using the `NA` identifier. `Y` can be omitted if the data object is used in a simulation.

The `Dose` component contains the bolus doses information. The elements used to describe a bolus dose are the time of dosing, the amount dosed and the state in the model into which the bolus is given. The `Time` vector in `Dose` contains the times to which doses are given. It is important that the time points in the `Dose` component is a subset of the overall `Time` vector otherwise the dose will not be given. The dose is given post-observation and prediction. This means that predictions to observations at times where a bolus dose is also given are formed prior to the "injection" of the dose. The elements `State` and `Amount` specifies in which compartment/state the dose should be given and what amount is given. Multiple doses are allowed at the same time point. Infusions can be specified using the input element u. The `covar` element contains the subject covariates (Z_i in Eq. (9)) and can be an array or list however the choice should be consistent with the referencing in the `h`function in Table 2.

3.3. *Package functionality*

Each of the three previously mentioned functionality parts is enclosed in a single function. The three functionality parts with corresponding functions are described in detail in the following sections.

3.3.1. *Simulation*

The function `PSM.simulate` performs the simulation of the system using an Euler based scheme. The simulation also includes inter-individual variation if the Ω matrix is specified. The stochastic noise term in the system equation is included by perturbing the states after each Euler step. The size of the pertubation is found by randomly sampling from a multivariate normal distribution with covariance proportional to the time step scaled with the diffusion scaling term. The default time step in the Euler scheme is 0.1 unless specified differently. It is upon the user to ensure that the observation times are a multiple of the time step.

The function arguments to `PSM.simulate` are as follows:

Table 4 – Simulation output.

Element	Type
Time	Vector
X	Matrix with the simulated states
Y	Matrix with the simulated observations
U	Matrix with the input used in the simulation
Dose	Dose list used in the simulation
eta	Matrix with the random effects used in the simulation
longX	Matrix with states at all sub-sampled timepoints
longY	Matrix with observations at all sub-sampled timepoints

```
PSM.simulate(Model, Data, THETA, deltaTime,
longX)
```

where Model and Data are lists as described in previous sections. Any observations in Data are disregarded as the simulation returns a set of observations Y. THETA is a vector with the population parameters to be used. deltaTime is the time step in the Euler scheme and the longX option is used to indicate whether the output should include all generated data at all sub-sampled time points or only return observations at prespecified time points.

The output from the simulation function is a list of lists where each underlying list contains the data for one subject. The list contains the elements shown in Table 4.

3.3.2. *Estimation*

The function PSM.estimate performs maximum likelihood estimation for the population parameters in the model. The objective function for the optimization is the negative log likelihood as defined in Eq. (11). The function calculates the numerical gradients and determines the optimal random effects needed in optimization of the likelihood function.

Functionality has been included to allow for constrained optimization using the logit transformation. A logit parameter transformation is used to convert the bounded parameters to unbounded parameters. In order to stabilize the optimization with boundaries a penalty function has also been included. The penalty increases as the parameter estimate approaches the boundary. The penalty function is introduced to ensure that the optimization will not get trapped in the flat regions of the logit transformation. For very large values in the unbounded parameter domain the transformed parameter will be close to the upper boundary. This also means that changing an extreme value in the unbounded parameter domain will hardly change the bounded parameter estimate. The optimizer stops when a change in the unbounded parameters does not change the likelihood function. The penalty function stabilizes this problem.

The currently used optimizer does not allow for NaN to be returned from the likelihood function. The search path for the optimization can lead to parameter values that generate NaN resulting in the search failing. This problem can be solved by using tighter boundaries and restarting the optimization with new boundaries in the recently found parameter values.

The parameter estimation based on the likelihood function consists of nested optimizations which makes the likelihood function highly nonlinear in parameters. The optimizer does not guaranty that the found minimum is the global minimum so the user should be aware of local minima and the importance of initial parameter values in the optimization. The user should preferably start the minimization in different initial parameter values to eliminate the risk of using parameter estimates from a local minimum in the modelling onwards.

To evaluate the quality of the parameter estimates the related uncertainties can be calculated. The uncertainties are based on the Hessian of the likelihood function. The parameter confidence bands are returned from the estimation procedure by setting the argument CI=TRUE. The Hessian is calculated using the numDeriv package.

The argument list for the estimation can be seen below:

```
PSM.estimate(Model, Data, Par, CI, trace, con-
trol, fast)
```

where the Model and Data are as previously described. Par is a list containing the initial parameter value in Init and optionally the upper and lower boundaries in UB and LB. CI specifies if the confidence intervals for the parameters should be calculated. trace is an integer controlling the amount of output from the optimization. The control argument is passed directly on to the optimiser—for further details see the help files for optim. The fast argument specifies whether the FORTRAN code should be used when possible. This can be useful for debugging purposes.

The Kalman Filter has been implemented in FORTRAN for linear models with non-singular system matrix. Non-linear models and singular linear models are implemented in R-code. The matrix exponential used for solving linear systems is also implemented in FORTRAN. Hence linear models with full matrices are estimated faster than other models. For initial modelling purposes it can often be extremely helpful to convert a singular model into a non-singular by adding a small rate constant to the diagonal.

The output from the estimation function can be seen in Table 5.

3.3.3. *Smoothing*

The estimation procedure relies on one-step predictions of observations based on previous data but in order to determine the overall most likely profile based on all data smoothing can be employed. The inclusion of all data allows noise effects occurring later in the time series to influence the profile earlier

Table 5 – Estimation output.

Element	Description
NegLogL	Negative log likelihood in the found optimum
THETA	Vector of parameters in optimum
CI	Confidence intervals based on the Hessian
SD	Standard error for the optimal parameters
COR	Correlation matrix for the optimal parameters
sec	Optimization time in seconds
opt	Messages from the optimizer

Table 6 – Smooth output.

Element	Description
`Time`	Sub-sampled time
`Xs`, `Ps`	Smoothed states and uncertainty
`Ys`	Predictions based on smoothed states
`Xf`, `Pf`	Filtered state and uncertainty
`Xp`, `Pp`	One-step state predictions and uncertainty
`Yp`, `R`	One-step observation predictions and uncertainty
`eta`	Estimated random effects
`NegLogL`	Negative log likelihood

on. Smoothing constructs the optimal state vector to all time points given both prior and future observations as opposed to filtered states that only depend on prior observations. The smoothed estimate is commonly used in post-processing of data as it represents the best fit based on all available data.

The function `PSM.smooth` performs smoothing of states using a Bryson Frazier algorithm [13].

The smoothing function argument list is shown below:

```
PSM.smooth(Model, Data, THETA, subsample, trace,
etaList)
```

where `Model` and `Data` are as described ealier. `THETA` is a vector with the population parameters for the evaluation, i.e. the returned parameter vector from the estimation. `subsample` is the number of sub-samples in between observations. Sub-sampling can be used to display the system behaviour in between observations. `trace` is an integer controlling the amount of text output. `etaList` is a matrix, where each column is the random effects for a subject. If `etaList` is set to `NULL` the random effects will be determined.

The complete listing of output from the smoother is shown in Table 6.

Several internal functions are used in simulation, estimation and smoothing functions and they can all be found in the documentation. For general use the three described functions will form a good base.

In the next section an application of the package is described.

4. Application: insulin secretion rates

In the article by Mortensen [7] the insulin secretion rate is determined by stochastic deconvolution using a Matlab framework. The insulin secretion rates are modelled as random walks and the Kalman Filter is used to determine the trajectory that most likely resulted in the observations. This section describes an extension implementing an intervention type model as known from Time Series Analysis in order to better characterize the insulin secretion.

The challenge in describing the insulin secretion is that the kinetic system for insulin is potentially non-linear and partly unknown. This makes insulin itself a poor descriptor for insulin secretion. During the production of insulin a by-product called connecting peptide (C-peptide) is produced in equimolar amounts. Insulin and C-peptide are split just as insulin is secreted into systemic circulation.

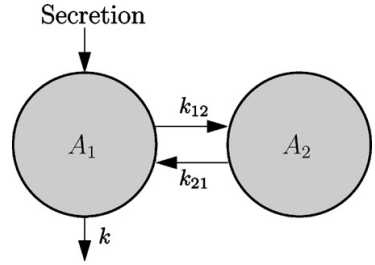


Fig. 2 – C-peptide PK model.

The pharmacokinetic system for C-peptide has been described in a population model by Van Cauter [14] with parameters based on subject covariates. The model structure is a linear two compartment model with elimination from the central compartment. The well known kinetics and longer half-life of C-peptide makes it a better descriptor of the actual insulin secretion even though the determined secretion rates are C-peptide secretion rates and not insulin secretion rates.

The graphical representation of the C-peptide model can be seen in Fig. 2. The exchange rate parameters and the elimination rate are all first order and the mathematical equations are given in (14)–(16):

$$\frac{dA_1}{dt} = SR - (k + k_{12})A_1 + k_{21}A_2 \tag{14}$$

$$\frac{dA_2}{dt} = k_{12}A_1 - k_{21}A_2 \tag{15}$$

$$y = \frac{A_1}{V} + e \tag{16}$$

where A_1 and A_2 are amounts in compartment 1 and 2. S R is the secretion rate measured as [amount/ min]. k, k_{12} and k_{21} are rate constants [min^{-1}]. V is the distribution volume [L].

The data originates from a meal tolerance test where the test subject is served three standardized meals over a period of 24 h. The insulin, C-peptide and glucose concentrations are measured throughout the 24 h and more frequently during meals. The C-peptide profiles can be seen in Fig. 3. The subjects are newly diagnosed type II diabetes patients with relatively intact insulin secretion. One complication with type II diabetes is decreasing beta cell mass, i.e. decreasing insulin secretion. To determine if the insulin secretion is intact, stochastic deconvolution [15] can be used.

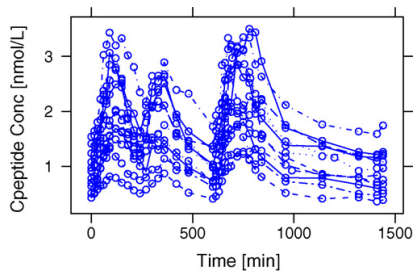


Fig. 3 – C-peptide profiles.

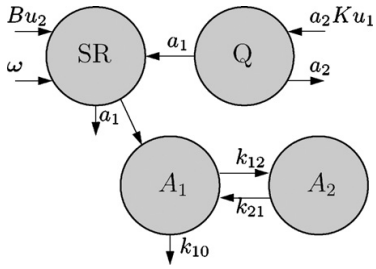


Fig. 4 – Intervention C-peptide model.

In the article [7] the insulin secretion is assumed to be a random walk and the most plausible trajectory is determined. The smoothed estimate is a compromise between the expected variation in the system and the observed variation in the observations. The scaling diffusion term $\boldsymbol{\sigma}$ is overestimated due to the physiological structure of the insulin secretion, where the secretion can only assume positive values. To remedy this assumption an intervention model is added to aid the structure of the secretion. The intervention model scheme is implemented by a step function located at meal time. A two compartment structure is assumed to allow for some flexibility in the form of the secretion rate profile. The secretion is furthermore assumed to have an underlying basal secretion rate. The basal rate, the rate parameters and the amplification are now estimated in this underlying structure for the secretion and the random walk is used to determine the deviations from this structure (Fig. 4).

The states of the system for further use in the equations are defined as below:

$$\mathbf{x} = [A_1, A_2, \mathrm{SR}, \mathrm{Q}]^{\mathrm{T}} \tag{17}$$

The mathematical equations describing the system can now be written as

$$\mathrm{d}\mathbf{x} = \left(\begin{bmatrix} -(k_{12}+k_{10}) & k_{21} & 1 & 0 \\ k_{12} & -k_{21} & 0 & 0 \\ 0 & 0 & -a_1 & a_1 \\ 0 & 0 & 0 & -a_2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & B\exp(\eta_2) \\ a_2 K \exp(\eta_1) & 0 \end{bmatrix} \boldsymbol{u} \right) \mathrm{dt} + \mathrm{diag} \begin{bmatrix} 0 \\ 0 \\ \sigma_{\mathrm{SR}} \\ 0 \end{bmatrix} \mathrm{d}\boldsymbol{\omega} \tag{18}$$

where k_x and a_x are rate constants. The input variable $\boldsymbol{u}_t = [u_1, u_2]'$ is used to model the baseline level of secretion (u_1) and the impulse effect from a meal (u_2). u_1 is equal to one for the entire time series whereas u_2 is only equal to one just after meals (i.e. from meal start and 30 min on). The B parameter specifies the baseline level in the secretion compartment and K specifies the amplitude of the meal impulse. Both B and K are modelled with an individual random effect (η_1, η_2). σ_{SR} is the scaling diffusion term for remaining description of the secretion rate.

The observation equation linking the model states to the C-peptide observations is shown below:

$$y = \left[\frac{1}{V}, 0, 0, 0\right] \mathbf{x} + e = \frac{A_1}{V} + e, \qquad \text{where} \quad e \in N(0, S) \tag{19}$$

This model is a simplification as the secretion responses to the meals are assumed equal over all three meals. An extension to make an individual secretion response per meal can be made by extending the input and the parameter list accordingly. This will however increase the number of parameters and thereby the estimation time.

Individual random effects have been added to the amplification of the response and the basal level so that each individual can have different secretion responses.

The parameters in the model to be estimated are the secretion parameters a_1, a_2, K, B and the variance parameter σ_{SR}. The inter-individual variance in $\boldsymbol{\Omega}$ is assumed to be 0.1. The residual variance is assumed to be $(50\,\mathrm{pmol/L})^2$ which was derived from the plot of the profiles.

4.1. *Model specification*

The model specification in `PSM` is described element by element in the next section.

```
> Cpep.Model <- list()
> Cpep.Model$h = function(eta,theta,covar) {
    theta$K12 <- covar[1]
    theta$K21 <- covar[2]
    theta$K10 <- covar[3]
    theta$V   <- covar[4]
    theta$K   <- theta$K*exp(eta[1])
    theta$B   <- theta$B*exp(eta[2])
    phi       <- theta }   }
```

Initially the model is setup by creating an empty list. The `h` function that translates the population parameters into individual parameters is specified. Function arguments that can be used in the creation of `phi` are population parameters, random effects and covariates. It can be seen that four individual parameters are extracted from the covariates and the two population parameters K and B are expanded with random effects.

```
> Cpep.Model$Matrices = function(phi) {
    K21  <- phi$K21  ;  K10 <- phi$K10
    K12  <- phi$K12  ;  a1  <- phi$a1
    a2   <- phi$a2   ;  V   <- phi$V
    matA <- matrix(
            c(-(K10+K12),K21,1,0,
```

```
        K12,-K21,0,0,
        0,0,-a1,a1,
        0,0,0,-a2),nrow=4,byrow=T)
matB <- matrix(
          c(0,0,0,a2*phi$K,
          0,0,phi$B,0), nrow=4)
matC <- matrix(c(1/V,0,0,0),nrow=1)
matD <- matrix(rep(0,2),nrow=1)
list(matA=matA,matB=matB,
     matC=matC,matD=matD)}
```

The first element created here is the time invariant matrices. In this example all four matrices need to be specified. First the individual parameters are extracted from `phi` and matrices are set up in a list named `Matrices`.

```
Cpep.Model$X0 = function(Time=Na,phi,U=Na) {
   K21  <- phi$K21  ;  K10 <- phi$K10
   K12  <- phi$K12
   B    <- phi$B    ;  a1  <- phi$a1
  matrix(c(B/(a1*K10),
           (B*K12)/(a1*K10*K21),
           B/a1,
           0),nrow=4) }
```

The initial conditions for the states are added to the list as an element named `X0`. The initial conditions used here are steady state conditions calculated using the basal secretion and kinetic parameters. The initial conditions are specified as a function with arguments `Time`, `phi` and `U`. The `Time` argument is the first time point specified and can be useful if the subjects start at different time points. The `U` can contain exogenous input to the system which enters into the initial conditions.

```
      Cpep.Model$SIG = function(phi) {
  SIG       <- matrix(rep(0,16),nrow=4)
  SIG[3,3]  <- phi$SIG33
  SIG }
Cpep.Model$S = function(phi) { matrix(phi$S)}
```

Table 7 – Estimated parameters.

Parameter	MLE	95% CI
K	1911.2	[1511.5; 2310.9]
B	7.019	[5.0; 9.037]
a1	0.029	[0.022; 0.036]
a2	0.011	[0.0089; 0.014]
SIG33	30.3	[26.05; 34.55]

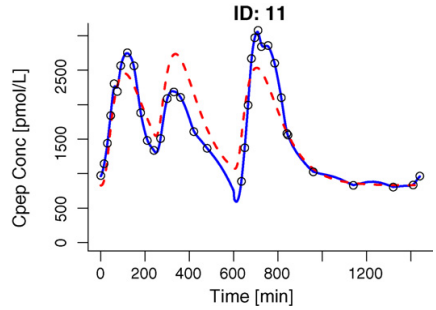


Fig. 5 – Fitted C-peptide concentrations for subject 11.

The diffusion scaling term and the residual variance are specified using the elements `SIG` and `S`. Both are specified as functions of `phi`. It can be seen that `SIG` is a 4×4 matrix with only an element at position $[3, 3]$. `S` is specified as a matrix even though it is one-dimensional.

```
> Cpep.Model$ModelPar = function(THETA){
  list(theta=list(K=THETA["K"],B=THETA["B"],
  a1=THETA["a1"],a2=THETA["a2"],S=50^2,
  SIG33=THETA["SIG33"]),
  OMEGA=diag(c(.1,.1))) }
```

The final element in the model is the function that splits the parameter vector containing parameters to be optimized into population parameters and inter individual covariance matrix $\boldsymbol{\Omega}$.

The list `Cpep.Model` now contains all the elements required in the model specification and the model can now be used to estimate parameters and create the smoothed profiles.

4.2. *Results*

The parameters in the C-peptide intervention model are estimated using `PSM.estimate` with constraints on the parameters. The optimal parameters and confidence bands are shown in the Table 7.

Two model fits for C-peptide can be seen in Figs. 5 and 6. The plots show the observations as circles and the intervention

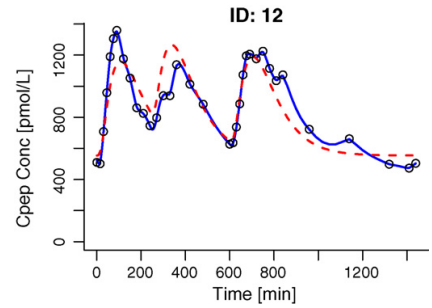


Fig. 6 – Fitted C-peptide concentrations for subject 12.

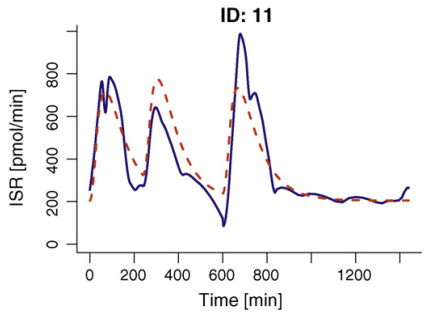


Fig. 7 – Secretion for subject 11.

model fit is shown in blue. The dashed line represents the deterministic model fit where the diffusion scale term has been fixed to zero in the intervention model in order to mimic an ODE model like in NONMEM. The dashed line is thus equal to a simulation with the intervention model.

The failure of the assumption of equal responses to all meals is obvious in the fit between the dashed red line and the observations. Large peaks are underestimated and small peaks overestimated. This is clear as the model describes the average response after a meal. The solution could be to extend the model so that every meal has it's own amplification.

As the model is used for deconvolution purposes the actual fit to the observations is of less importance, but more interesting is the secretion rate profile and the split between the model and the Wiener component.

The two corresponding secretion rate profiles to the C-peptide fits are found in Figs. 7 and 8. The full line is the optimal secretion rate determined by the Kalman Filter and the dashed line represents the deterministic part of the model. The secretion model captures the overall trends but the compromise with the equal response assumption is clear.

The deconvoluted insulin secretion rates have some jumps which seem unphysiological. The solution could be to use an integrated random walk as driver for insulin secretion. This would make the deconvoluted insulin secretion rate less erratic.

This section has shown a simple application of the PSM package for modelling purposes. Classic log likelihood ratio testing can also be applied in the model development as well as visual predictive checks of the model fits.

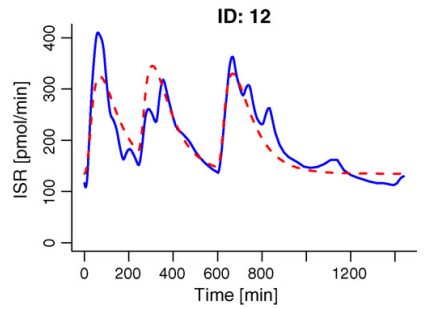


Fig. 8 – Secretion for subject 12.

5. Discussion and conclusion

The PSM package provides a new general framework for handling dynamic models based on stochastic differential equations in a population setting. The package is available in R to ease availability, ease installation and enable a single working environment for data handling, modelling and visualization.

The package is an extension to the Matlab framework of the same name. The package functionality is a combination of the functionality available in CTSM and NONMEM. CTSM and NONMEM have also been used to validate the package as described further on the homepage [16].

The package enables the feature of dosing capabilities which makes the package useful in PK/PD model development. This is further supported by the ability to handle multidimensional observations which aids in modelling work with observations from both PK and PD.

The package includes functionality for modelling tasks such as simulation, parameter estimation and smoothing. The stochastic deconvolution example with an underlying secretion model in this article showed an application of the package.

The computational effort in working with larger models is substantial and the use of parallelization could decrease the time considerably in the modelling. It is currently being investigated how to implement parallelization in a general manner in R. Parallelization is an obvious solution due to the fact that future computer systems will be massive multicore systems.

The package is a promising tool to get started with stochastic differential equations or the inclusion of mixed-effects in continuous-discrete state-space models.

More information can be found on the webpage http://www.imm.dtu.dk/psm.

Acknowledgement

This work was supported by the European Union through the Network of Excellence BioSim, contract no. LSHB-CT-2004-005137.

REFERENCES

[1] N.R. Kristensen, H. Madsen, S.H. Ingwersen, Using stochastic differential equations for PK/PD model development, J. Pharmacokinet. Pharmacodyn. 32 (February(1)) (2005) 109–141.

[2] C.W. Tornøe, R.V. Overgaard, H. Agersø, H.A. Nielsen, H. Madsen, E.N. Jonsson, Stochastic differential equations in NONMEM: implementation, application, and comparison with ordinary differential equations, Pharm. Res. 22 (August(8)) (2005) 1247–1258.

[3] R.V. Overgaard, N. Jonsson, C.W. Tornøe, H. Madsen, Non-linear mixed-effects models with stochastic differential equations: implementation of an estimation algorithm, J. Pharmacokinet. Pharmacodyn. 32 (February(1)) (2005) 85–107.

[4] U. Picchini, S. Ditlevsen, A. De Gaetano, Maximum likelihood estimation of a time-inhomogeneous stochastic differential

model of glucose dynamics, Math. Med. Biol. 25 (June(2)) (2008) 141–155.
[5] L.B. Sheiner, S.L. Beal, NONMEM User's Guide, NONMEM Project Group, University of California, San Francisco, 1994.
[6] N.R. Kristensen, CTSM Mathematics Guide, Technical Report, Technical University of Denmark, December 2003.
[7] S.B. Mortensen, S. Klim, B. Dammann, N.R. Kristensen, H. Madsen, R.V. Overgaard, A Matlab framework for estimation of NLME models using stochastic differential equations: applications for estimation of insulin secretion rates, J. Pharmacokinet. Pharmacodyn. 34 (October(5)) (2007) 623–642.
[8] C.W. Tornøe, H. Agersø, E. Niclas Jonsson, H. Madsen, H.A. Nielsen, Non-linear mixed-effects pharmacokinetic/pharmacodynamic modelling in NLME using differential equations, Comput. Meth. Programs Biomed. 76 (October(1)) (2004) 31–40.
[9] R.E. Kalman, New approach to linear filtering and prediction problems, Am. Soc. Mech. Eng. – Trans. – J. Basic Eng. Ser. D 82 (1) (1960) 35–45.
[10] G. Welch, G. Bishop, http://www.cs.unc.edu/~welch/kalman/.
[11] R.V. Overgaard, Pharmacokinetic/Pharmacodynamic Modeling with a Stochastic Perspective. Insulin Secretion and Interleukin-21 Development as Case Studies, PhD Thesis, Technical University of Denmark, Informatics and Mathematical Modelling, 2006.
[12] L.B. Sheiner, S.L. Beal, Evaluation of methods for estimating population pharmacokinetics parameters. I. Michaelis–Menten model: routine clinical pharmacokinetic data, J. Pharmacokinet. Biopharm. 8 (December(6)) (1980) 553–571.
[13] T. Kailath, A.H. Sayed, B. Hassibi, Linear Estimation, Prentice-Hall, 2000.
[14] E. Van Cauter, F. Mestrez, J. Sturis, K.S. Polonsky, Estimation of insulin secretion rates from c-peptide levels comparison of individual and standard kinetic parameters for c-peptide clearance, Diabetes 41 (March(3)) (1992) 368–377.
[15] N.R. Kristensen, A deconvolution method for linear and nonlinear systems based on stochastic differential equations, Population Approach Group Europe, 2004.
[16] S. Klim, S.B. Mortensen, PSM homepage. Internet: http://www.imm.dtu.dk/psm, June 2008.

Appendix D

Paper D

Title:
Local estimation of a discretely observed continuous time inhomogeneous markov jump process.

Authors:
S. B. Mortensen, H. Madsen, and P. Hougaard.

Submitted to:
Statistical Modelling (May 2009).

Local Estimation of a Discretely Observed Continuous Time Inhomogeneous Markov Jump Process

Authors: Stig B. Mortensen[1,2,†], Henrik Madsen[1] and Philip Hougaard[2]

[1] DTU-Informatics, Technical University of Denmark, Lyngby, Denmark
[2] H. Lundbeck A/S, Valby, Denmark
[†] Corresponding author: sbm@imm.dtu.dk

May 14, 2009

Abstract

A general estimation method for a discretely observed inhomogeneous continuous time Markov jump process is presented. The method is able to handle time series data with a discrete sample space, where it is reasonable to assume that the data is generated by a slowly varying inhomogeneous Markov jump process. The estimation method uses local kernel estimation in combination with a weighted log-likelihood model to capture time variations of the Markov process. An application of the method is illustrated on data of discretely sampled sleep stages from an EEG study on rats.

Keywords:
Kernel estimation, maximum likelihood, Markov process, sleep EEG data.

Sumitted for:
Statistical modelling (May 14, 2009).

1 Introduction

In many applications one may encounter discretely sampled data that has a finite sample space. Examples of such are financial data with credit ratings, observations of cloud cover or observations of sleep stages during human or animal sleep. A first assumption for such data will often be that the data is generated by a time homogeneous Markov process. The homogeneous Markov assumption is often approximately valid over short time intervals, but if data is sampled over longer time periods this may no longer be the case due to changes over time of the system. Relating to the above mentioned examples the cause could be new financial market conditions, changes of season and changes in the sleep process.

One way to handle such changes in the dynamics of the process is to apply an inhomogeneous continuous time Markov model. This is an attractive model as it has a simple model structure but is still very flexible. Being inhomogeneous the model is able to reflect changes in the system and by further formulating the model in continuous time will often allow a more simple parameterization of the model that can be related to physical constraints in the system. This paper suggests a novel estimation procedure that based on discrete measurements will estimate changes in the model parameters over time for the continuous Markov process.

The theory of the method is presented in Section 2 and 3 and in Section 4 its application is illustrated with data from a study of EEG sleep stages in rats.

2 Model

Let $\{X(t) \in S; t \in T\}$ be a continuous time inhomogeneous Markov process with a finite state space $S = \{1, ..., m\}$ and $T = [0, t_{max}]$. The process is defined by a stochastic matrix of transition probabilities $\boldsymbol{P}(t,u) = \{p_{ij}(t,u)\}$ with

$$p_{ij}(t,u) = \Pr\{X(u) = j | X(t) = i\}. \tag{2.1}$$

By introducing the corresponding intensity matrix $\boldsymbol{Q}(t) = \{q_{ij}(t)\}$ it can be shown that

$$\frac{\partial}{\partial u}\boldsymbol{P}(t,u) = \boldsymbol{P}(t,u)\boldsymbol{Q}(u), \quad u,t \in T, \tag{2.2}$$

which is known as the Kolmogorov forward differential equation. The equation shows that the process is also defined completely through the $\boldsymbol{Q}(t)$ matrix (Cox and Miller, 1965, p. 181). The $\boldsymbol{Q}(t)$ matrix has diagonal elements $q_{ii} \leq 0$ and off-diagonal elements $q_{ij} \geq 0$, $i \neq j$, and the matrix will always be defined given a Markov process with a stochastic matrix $\boldsymbol{P}(t,u)$. It can be shown that if $\boldsymbol{Q}(t)$ is time invariant then times between jumps (holding times) are exponentially distributed and the diagonal elements $q_{ii}(t)$ of $\boldsymbol{Q}(t)$ contains the negative rate

for leaving a state. The fraction $-q_{ij}(t)/q_{ii}(t)$ is the probability of going to state j when a jump from state i occurs. Consequently, the row sum of $\boldsymbol{Q}(t)$ is 0 since $\sum_{j\neq i} -q_{ij}/q_{ii}$ must be 1, meaning that the process must jump to a state in S with probability one when a transition occurs.

The process $X(t)$ is observed at N discrete time points that are chosen independent of the observed process. The model dynamics are assumed to be slowly varying relative to the time between observations. The observation sequence is denoted $\{x_k\}$ and contains the state at time t_k where $k = 1, ..., N$. There is no requirement of an equidistant sampling scheme, and therefore "missing" observations do not need to be handled separately.

3 Estimation

Given a set of observations we wish to find an estimate of $\boldsymbol{Q}(t_c)$ at a time point $t_c \in T$ using a locally-weighted maximum likelihood frame work. To further estimate the time variations of $\boldsymbol{Q}(t)$ over the entire interval T one simply applies the local model sequentially for a series of suitably close time points in T. The presentation of the estimation method is thus only concerned with the point estimate $\boldsymbol{Q}(t_c)$ the remaining part of the section. The dependence on the chosen time point t_c is suppressed in the notation to ease readability, but it is important and should be kept in mind.

It will often be the case that the estimation of $\boldsymbol{Q}(t)$ should be limited to a particular structure of interest. For this purpose we define in $\boldsymbol{f}$ a parameterization for the $\boldsymbol{Q}(t)$ matrix with a set of parameters $\boldsymbol{\theta}(t) = (\theta_1(t), \ ... \ , \theta_v(t))$ such that

$$\boldsymbol{f}: \quad \boldsymbol{\theta}(t) \quad \rightarrow \quad \boldsymbol{Q}(t). \tag{3.1}$$

The parameterization in (3.1) can be used to define a particular structure of $\boldsymbol{Q}(t)$ based on a knowledge of the process. Thus instead of estimating the full $\boldsymbol{Q}(t)$ matrix it is only nescessary to estimate the parameters representing possible jumps in continuous time. This typically limits the number of parameters in $\boldsymbol{Q}(t)$ that has to be estimated. The parameterization in (3.1) can also be used to define a more suitable transformation of the elements in $\boldsymbol{Q}(t)$ to be used in the maximum likelihood estimation. The elements $\{q_{ij}(t)\}$ are all rate related parameters and it may thus be desirable to estimate the logarithm of these to make the parameter space unbounded.

Next, define in $\boldsymbol{g}$ a series of local polynomial models of order r for the parameters $(\theta_1(t), \ ... \ , \theta_v(t))$ such that

$$\boldsymbol{g}: \quad (\boldsymbol{\beta}, t) \quad \rightarrow \quad \boldsymbol{\theta}(t) \tag{3.2}$$

where

$$\theta_s(t) = \beta_{s0} + (t - t_c)\beta_{s1} + \ ... \ + (t - t_c)^r \beta_{sr} \tag{3.3}$$

with the matrix $\boldsymbol{\beta} = \{\, \beta_{kl} \,\}$, $s = 1, ..., v$ and $l = 0, ..., r$. It then holds that a local estimate of the $\boldsymbol{Q}(t_c)$ matrix is given by $\hat{\boldsymbol{Q}}(t_c) = \boldsymbol{f}(\boldsymbol{g}(\hat{\boldsymbol{\beta}}, t_c))$, where $\hat{\boldsymbol{\beta}}$ is an estimate of $\boldsymbol{\beta}$. This estimate will be based on a locally-weighted likelihood function, which will be described in the following.

First we define a series of stochastic matrices

$$\boldsymbol{P}_k = \{p_{ij}\}_k = \{p_{ij}(t_k, t_{k+1})\} \tag{3.4}$$

containing the probabilities for observing $x_{k+1} = j$ given $x_k = i$. These transition probabilities are found by solving (2.2) with initial condition $\boldsymbol{P}(t_k, t_k) = \boldsymbol{I}$ and using $\boldsymbol{Q}(u) = \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{\beta}, u))$. The solution can be found numerically using standard algorithms for solving ordinary differential equations. Alternatively, if the time interval $\Delta t_k = t_{k+1} - t_k$ is small, one may as an approximation use a zero or first order Taylor expansion of $\boldsymbol{Q}(t)$ around t_k. The most simple and in many cases sufficient approximation is the zeroth order expansion where the intensity matrix is assumed constant in each time interval t_k to t_{k+1}. In this case the solution to (2.2) is given explicitly as

$$\begin{aligned} \boldsymbol{P}_k &= \exp[\, \boldsymbol{Q}(t_k)\, \Delta t_k \,] \\ &= \exp[\, \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{\beta}, t_k))\, \Delta t_k \,] \end{aligned} \tag{3.5}$$

where $\exp[\cdot]$ represents the matrix exponential function. This approximation is computationally faster, and will in many cases be sufficiently accurate as long as the time variations in $\boldsymbol{Q}(t)$ are small compared to the time interval between consecutive observations.

In order to simplify the description of the likelihood function for $\boldsymbol{\beta}$ an indicator observation $\{y_{ij}\}_k$ is defined from x_k with $i, j = 1, ..., m$ and $k = 1, ..., N-1$. Let $y_{ijk} = 1$ when $x_k = i$ and $x_{k+1} = j$ and otherwise $y_{ijk} = 0$. The observation y_{ijk} thus indicates if a change of state from i to j is observed from time t_k to t_{k+1}. This new set of indicator observations contains all the information in the original set of observations $\{x_k\}$ and is thus also a sufficient statistic for estimating $\boldsymbol{\beta}$. Using the transition probabilities defined in (3.4) we know that the likelihood of observing $x_k = i$ followed by $x_{k+1} = j$ is p_{ijk}. The contribution to the log-likelihood for one observation y_{ijk} can thus be written as $y_{ijk} \log p_{ijk}$ and combining all m^2 indicator observations at time t_k this gives

$$\log L_k(\boldsymbol{\beta}) = \sum_i \sum_j y_{ijk} \log p_{ijk}(\boldsymbol{\beta}). \tag{3.6}$$

We now define weights for the observations as $w_k(t_c) = K_h(t_k - t_c)$ using a symmetric probability function K (also often referred to as a kernel function). The parameter h is the bandwidth controlling the size of the local neighborhood such that $K_h(t) = K(t/h)/h$ (Fan and Gijbels, 1996, p. 15). This can be used

to form a locally weighted log-likelihood function for $\boldsymbol{\beta}$ as

$$\begin{aligned} \log L(\boldsymbol{\beta}, t_c) &= \sum_k w_k(t_c) \log L_k(\boldsymbol{\beta}) && (3.7) \\ &= \sum_k \sum_i \sum_j w_k(t_c)\, y_{ijk} \log p_{ijk}(\boldsymbol{\beta}) && (3.8) \end{aligned}$$

and $\boldsymbol{\beta}$ may thus for a given time point t_c be estimated locally as

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \log L(\boldsymbol{\beta}, t_c). \tag{3.9}$$

Given that the local polynomia in (3.3) are centered around t_c, a local estimate of each parameter $\theta_s(t_c)$ for a given t_c is directly β_{s0}. The estimate of all parameters in $\hat{\boldsymbol{Q}}(t_c)$ is thus given as

$$\begin{aligned} \hat{\boldsymbol{\theta}}(t_c) &= \boldsymbol{g}(\hat{\boldsymbol{\beta}}, t_c) = \hat{\boldsymbol{\beta}}_0 \\ &= [\ \hat{\beta}_{10},\ \hat{\beta}_{20},\ \ldots\ , \hat{\beta}_{v0}\ ]. \end{aligned}$$

3.1 Local constant model

The estimation method can be simplified if sample times are equidistant with $t_k = t_1 + (k-1)\Delta t_k$ and if it is chosen to use a local constant model instead of a higher order polynomial.

Thus, if we in (3.3) let $r = 0$ such that $\theta_s(t) = \beta_{s0}$ then $\boldsymbol{Q}(t)$ modelled as $\boldsymbol{f}(\boldsymbol{g}(\boldsymbol{\beta}, t))$ will be constant with respect to t for a given $\boldsymbol{\beta}$ and the transition probabilities can be found using the matrix exponential in (3.5) without approximation. Note that when using a local constant model in (3.3) the estimate $\hat{\boldsymbol{Q}}(t_c)$ is still varying with respect to t_c due to the weighing of the observations that depend on t_c. Assuming equidistant sampling times then also Δt_k will be constant, and this results in time invariant transition probabilities for a given $\boldsymbol{\beta}$. It is therefore only necessary to solve (2.2) one time instead of $N-1$ times to evaluate the likelihood function for $\boldsymbol{\beta}$ and we may write the transition probabilities $p_{ijk}(\boldsymbol{\beta})$ as $p_{ij}(\boldsymbol{\beta})$ for any k to emphasize this.

By defining a "weighted" number of jumps from i to j as $n'_{ij} = \sum_k w_k(t_c)\, y_{ijs}$ the locally-weighted log-likelihood function in (3.8) can be written as

$$\begin{aligned} \log L(\boldsymbol{\beta}, t_c) &= \textstyle\sum_k \sum_i \sum_j w_k(t_c)\, y_{ijk}\ \log p_{ijk}(\boldsymbol{\beta}) \\ &= \textstyle\sum_i \sum_j \log p_{ij}(\boldsymbol{\beta}) \sum_k w_k(t_c)\ y_{ijs} \\ &= \textstyle\sum_i \sum_j \log p_{ij}(\boldsymbol{\beta})\, n'_{ij}, \end{aligned} \tag{3.10}$$

where the sum over all observations is removed from the log-likelihood function. This formulation of the estimation problem gives a significant computational

advantage. The matrix $\{n'_{ij}\}$ can be pre-computed for a given t_c since it does not depend on $\boldsymbol{\beta}$ and an evaluation of the log-likelihood function then only requires solving (2.2) once to compute $\{p_{ij}(\boldsymbol{\beta})\}$. This not only reduces time required for estimation, but also improves the numeric stability of the log-likelihood evaluation.

3.2 Bias vs. variance of estimate

The local estimation method provides a point estimate of the parameters $\boldsymbol{\theta}(t_c)$ defining $\boldsymbol{Q}(t_c)$ via the parameterization given in $\boldsymbol{f}$. For any local estimation procedure there will be a trade-off between reducing bias and variance of this point estimate.

To find an estimate of the variance we may rely on the asymptotic distribution theory of the maximum likelihood estimator, since the local estimation method is based on a likelihood function (Pawitan, 2001). The observed Fisher information for $\hat{\boldsymbol{\beta}}_0(t_c)$ is

$$\boldsymbol{I}_0(t_c) = -\frac{\partial^2}{\partial \boldsymbol{\beta}_0^2} \log L(\boldsymbol{\beta}, t_c)\Big|_{\boldsymbol{\beta}_0=\hat{\boldsymbol{\beta}}_0} \tag{3.11}$$

and using that $\hat{\boldsymbol{\theta}}(t_c) = \hat{\boldsymbol{\beta}}_0(t_c)$ it thus holds asymptotically that

$$\text{var } \hat{\boldsymbol{\theta}}(t_c) = \boldsymbol{I}_0^{-1}(t_c). \tag{3.12}$$

if $\hat{\boldsymbol{\theta}}(t_c)$ is unbiased. From likelihood theory it is known that the variance estimate in (3.12) will decrease for an increasing bandwidth, since this will include more observations in the likelihood function. However, $\hat{\boldsymbol{\theta}}(t_c)$ will only be unbiased if the local model is correct on a global level, and this can generally not be assumed. The polynomial model is chosen because it is well suited for locally modelling a true function of any any shape. If the bandwidth is increased to much this will no longer be the case and the decreasing variance will come at the cost of increasing bias. The bandwidth should thus not be increased beyond a range where the polynomial approximation is expected to be sufficient and the variance estimate in (3.12) should only be seen as a an indication of the precision of the estimate in situations where bias is expected to be relatively small.

For local estimation in a least square setting (also known as non-parametric regression) with a model of the type $z_k = g(t_k) + \epsilon_k$ the problem of balancing bias and variance in estimation of the unknown regression function g has a long history. Classical methods for finding an optimal bandwidth include cross validation criterion and related methods (Wahba and Wold, 1975; Rice, 1984; Müller, 1985; Müller et al., 1987). With cross validation we define a cross validation function $CV(\lambda) = \frac{1}{n}\sum_k \left(z_k - \hat{g}^*_\lambda(t_k)\right)^2$ where $\hat{g}^*_\lambda(t_k)$ is the non-parametric estimate of $g(t_k)$ using all observations except observation z_k. The parameter

λ is the reciprocal of the bandwidth, and minimizing the cross validation function with respect to λ gives an objective choice of the optimal bandwidth. The cross validation criterion can be generalized to other types of responses using the theory of generalized linear models where deviance is introduced to measure agreement between observations and model (McCullagh and Nelder, 1983). If we assume a Gaussian distribution of ϵ_k and use $\mu_k = g(t_k)$ the deviance for one observation is defined as $D(z_k, \mu_k) = (z_k - \mu_k)^2$. In this way the cross validation function can be defined as $CV(\lambda) = \frac{1}{n}\sum_k D(z_k, \hat{\mu}_k^*)$ where D is the individual deviance contribution for observation z_k and $\hat{\mu}_k^* = \hat{g}_\lambda^*(t_k)$. In this form the cross validation technique can be extended to all types of responses in the exponential family distribution (O'Sullivan et al., 1986) and this also includes the binary observations y_{ijk} of a Markov process presented here.

For binary observations it can be shown that the deviance is equal to minus 2 log-likelihood of the binary observation. For the indicator observations y_{ijk} we are only interested in the deviance contribution when $y_{ijk} = 1$ since this indicates an actual observation. The deviance can thus be defined as

$$D(y_{ijk}, p_{ijk}) = -2y_{ijk} \log p_{ijk}.$$

This can be used for defining a cross validation criterion as

$$CV(\lambda) = \frac{2}{n} \sum_i \sum_j \sum_k -y_{ijk} \log \hat{p}_{ijk}^*$$

where $\hat{p}_{ijk}^*$ is found using a bandwidth $h = 1/\lambda$ and excluding observation y_{ijk}. The cross validation criterion gives an objective measure for the optimal bandwidth in the local Markov model. The criterion will however quickly become computationally demanding to evaluate with increasing number of observations, since it requires a local estimate $\hat{p}_{ijk}^*$ at each time point t_k to evaluate $CV(\lambda)$. In practical situations it is thus only feasible to compute if the number of observations is relatively small.

In problems where the number of observations is large as in the example presented in Section 4 other approaches are needed. Instead the situation can be seen in analogy to estimation of the spectral density function in time series analysis, where it is well known that some degree of smoothing is nescessary to achieve a consistent estimate (Priestley, 1981). The degree of smoothing is chosen such that it seems to reasonably balance bias and variance. The same approach is suggested here, that is simply to redo the local estimation method for a range of bandwidth values to find the most suitable value.

3.3 Type of bandwidth and kernel function

The type of bandwidth and kernel function K is also important in order to achieve the best performance of the local estimation. There are a number of

commonly used kernel functions available, such as the Gaussian kernel, tricube kernel and the uniform, Epanechnikov, biweight and triweight kernels that are derived from the symmetric Beta family (Fan and Gijbels, 1996). A number of the kernels are shown in Figure 1. The Gaussian kernel has the disadvantage of unbounded support and hence positive weights for all data points. This makes the local estimation procedure computationally more demanding, since all data has to be considered at any chosen time point t_c. The remaining kernel functions have only bounded support in the interval $[-1; 1]$ which is mapped to the interval $[-h; h]$ in the scaled kernel function K_h. The tricube kernel is often preferred because it approaches its bounds smoothly.

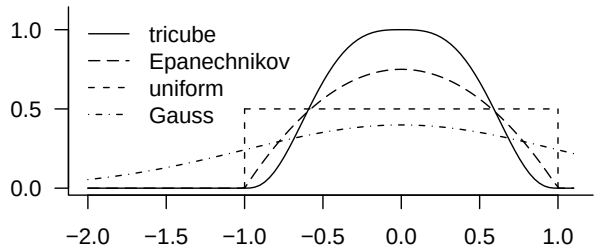


Figure 1: Kernel functions.

As discussed in the previous section a smaller bandwidth reduces estimation bias but increases the variance and vice versa. Two types of bandwidth have been considered here.

The first type of bandwidth is to use a common fixed bandwidth h. This is the most simple way of specifying the bandwidth. It can be useful in many cases, but it can give large differences in the precision of the parameter estimates over time. This happens if the observations are sampled with varying frequency and thereby results in varying numbers of observations in the fixed length intervals $t_c + [-h; h]$. The fixed bandwidth h must naturally be positive, but it is not upwards bounded as increasing h just results in an increasingly smooth estimate of the time variations.

The second choice for a bandwidth is to choose a data dependent bandwidth h_{t_c} such that the interval $t_c + [-h_{t_c}; h_{t_c}]$ always includes a fixed proportion α of the total number of observations. This is often denoted a nearest neighbor (NN) bandwidth. The definition can be extended to allow $\alpha > 1$ by scaling h_{t_c} by $\max(1, \alpha)$. The nearest neighbor bandwidth will help control estimation variance when the frequency of observations varies.

If sample times are equidistant the two types of bandwidths are almost equivalent, since a fixed time window contains a fixed proportion of the data. They will only be different near the boundary, where the data dependent bandwidth increases to contain the fixed proportion of the data. It is noted that for both

choices of bandwidth, when $h \to \infty$ or $\alpha \to \infty$ the local estimation will approach a global polynomial estimation.

There are further aspects that can be considered when choosing kernels and bandwidth. For local estimation in a least square setting the estimation near the boundary (when $t_c < h$ or $t_c > t_{max} - h$) can give problems with increased bias. A common way to avoid this is to use special boundary kernels, see e.g. Gasser et al. (1985). Similar approaches could also be considered in this context to improve the estimate at the boundary.

A special issue that in only relevant for the local estimation presented here is that a chosen bandwidth must contain observations of all states in order to estimate all parameters in the model. If a particular state in the process is observed much less frequently than other states, this will set a lower bound on the possible bandwidth. A way to give more robust local estimates is thus to use a state dependent bandwidth such that e.g. a proportion α of the observations of *each state* is included in the estimation. This can be done by using state dependent weights $w_{ik}(t_c)$ instead of $w_k(t_c)$ in (3.8) where $w_{ik}(t_c)$ is found with a different bandwidth depending on from which state the observed jump y_{ijk} occurs.

4 Example

To illustrate the use of modelling based on local estimation of a continuous time inhomogeneous Markov process, the method will be applied to data from a study monitoring the sleep cycle of rats.

The data has been collected in a pre-clinical trial using 6 rats that have been observed for 23.5 hours (Anderson et al., 2005). At the beginning of the period the rats were given an oral dose of 20mg Gaboxadol, which is a sleep promoting compound. Previous experience has indicated that the drug is present in the brain for about 8 hours and reaches the highest concentration after approximately 1 hour. During the first 12 hours the lights were on, and in the remaining time the lights were off. Since rats are most active in the dark, this study design is meant to resemble a human taking a sleep drug before bed time.

The sleep cycle is monitored using electroencephalography (EEG) which is used to determine the sleep stage of the rat. Three states are classified, namely wake (W), delta sleep (DS) and paradoxical sleep (PS) and the states are numbered in that order. These states are determined every 10 seconds giving 8,460 equidistantly spaced observations for each rat. In order to simplify the example it is chosen to regard the observations from all six rats as realizations of the same Markov process. The local estimation thus reflects an average individual process.

The implementation of the estimation method has been done in R (R Development Core Team, 2008) and it was chosen to use the matrix exponential in (3.5) to solve (2.2) since data is sampled at high frequency compared to the expected variations it $\boldsymbol{Q}(t)$. The matrix exponential is not part of base R so it was chosen to use the Fortran routine `DGPADM` from EXPOKIT (Sidje, 1998) for this purpose.

The parameterization $\boldsymbol{f}$ of the $\boldsymbol{Q}(t)$ matrix is shown in (4.1). The structure is based on Madsen et al. (1985), where $\boldsymbol{Q}(t)$ is parameterized with two parameters for each state, one rate parameter for leaving the state and one parameter for the probability of changing one state down when a jump occurs. This parameterization is advantageous if it is only possible to jump to neighboring states, since the number of parameters only increases linearly with the number of states instead of quadratically when using a full matrix. In this example we will use the model

$$\begin{aligned} \boldsymbol{f} : \boldsymbol{\theta}(t) &\rightarrow \boldsymbol{Q}(t) : \\ \boldsymbol{Q}(t) &= \begin{bmatrix} -q_1(t) & q_1(t) & 0 \\ w_2(t)q_2(t) & -q_2(t) & (1-w_2(t))q_2(t) \\ (1-w_3(t))q_3(t) & w_3(t)q_3(t) & -q_3(t) \end{bmatrix} \\ q_i(t) &= \exp[\theta_i(t)], \quad i = 1,2,3 \\ w_i(t) &= \text{logit}^{-1}[\theta_{i+2}(t)], \quad i = 2,3\,, \end{aligned} \tag{4.1}$$

which is illustrated in Figure 2. Since the rats have only three states, the structure with two parameters per state does not reduce the number of parameters needed, but it is still used to illustrate the use of the approach.

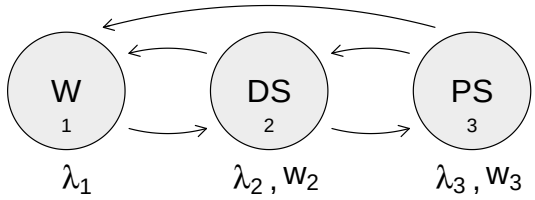


Figure 2: Model for sleep stage transitions.

In the parameterization in (4.1) we furthermore use the constraint that $w_1(t) = 0$ meaning that jumps from W to PS should never occur. This reflects that it is known from physiology that the PS state is always preceded by the DS state. There is a very small number of observations in the data where W is followed by an observation of PS, but by using $w_1(t) = 0$ we thereby assume that the process has been in the DS state in between observation times.

For the local estimation we use a second order polynomial in $\boldsymbol{g}$ and a tricube kernel with a nearest neighbor bandwidth of $\alpha = 0.40$. The result of the estimation is shown in Figure 3 together with Wald 95% confidence intervals found using (3.12).

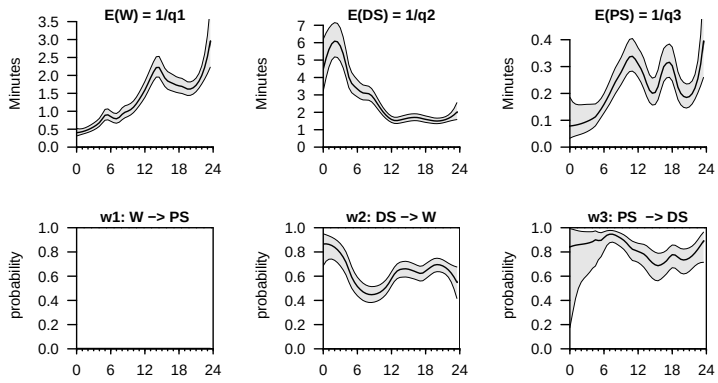
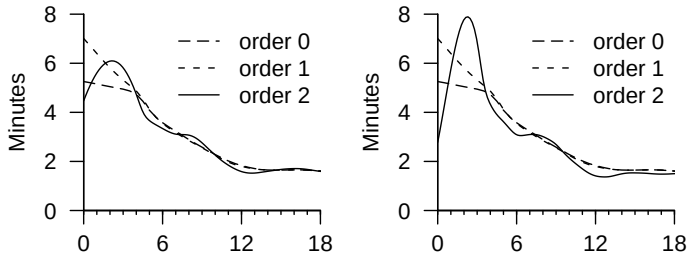


Figure 3: Estimate of the $\boldsymbol{Q}(t)$ matrix as a function of time using 2nd order local polynomials and a NN bandwidth $\alpha = 0.40$. The surrounding lines are Wald 95% pointwise confidence intervals.

The first row in Figure 3 shows the expected time in each state before leaving (holding times), which is found as $1/q_i(t)$ since the time in each state is approximately exponentially distributed. Looking at holding times is chosen here since it can be a more intuitive way to view the rate parameters, but it is only valid if the holding times are short compared to the time variations of the parameters as discussed previously. It is seen that the estimate of time in W increases after 12 hours (lights off) along with the probability of jumping from DS to W when leaving DS, and this seems reasonable as the activity should increase in the dark period. The expected time in DS reflects the influence of the drug, as it peaks at about three hours, which indicates a delay of 2 hours with respect to the peak of the drug concentration. The most notable effect on the pointwise confidence intervals is seen for w_3 between 0 and 6 hours. This reflects that there is only limited information about the probability of going from PS to DS or W, since the process only rarely enters the PS state in this period as seen in the figure for $1/q_3$.

In the example shown in Figure 3 the estimation is done using a second order polynomial with an NN bandwidth $\alpha = 0.40$. To demonstrate the effect of using lower orders, this is done for the DS state and shown in Figure 4(a) for the interval 0 to 18 hours. Both the zero and first order approximations are expected to give negative bias around peaks, and this is seen clearly from the figure around the peak at 2-3 hours. For the later times in the interval 6 to 18 hours the three approximations perform equally well. In Figure 4(b) the bandwidth is further reduced to $\alpha = 0.30$. This gives increased sensitivity and the second order local model is here seen to capture a stronger peak, which further strengthen the evidence for a true significant increase in the expected time in DS is present. It has been tried to use an even lower bandwidth, but unfortunately the estimation of especially w_3 becomes unstable. This is due

to a very limited number of observations of the PS state during the first hours of the study, and for lower choices of bandwidth the estimation was no longer possible. To further reduce the bandwidth it is nescessary to consider a state dependent bandwidth as mentioned in Section 3.3.



(a) $\alpha = 40\%$ (b) $\alpha = 30\%$

Figure 4: Expected time in DS state for different order of local polynomial using two choices of NN bandwidths.

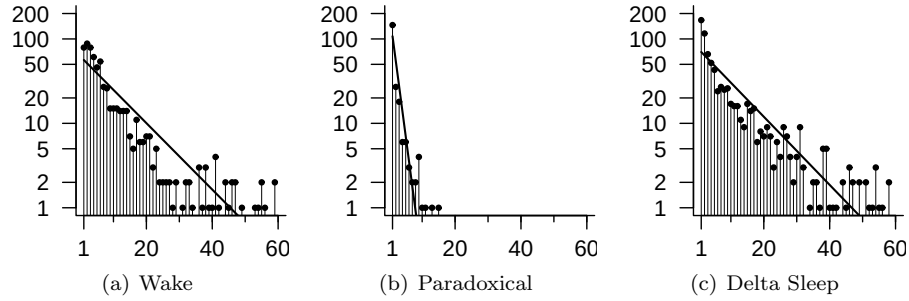
It is also of interest to get an impression of whether the Markov assumption is valid. As noted earlier the holding times should be exponentially distributed when the Markov process is homogeneous. Based on Figure 3 it can be seen that the process can be assumed to be roughly homogeneous in the time between 14 and 22 hours. Since data are sampled equidistantly the distribution of the observed holding time follows a geometric distribution. It is chosen to look at the run lengths as this is proportional to the observed holding times. The distribution of the run lengths K is given as

$$\Pr(K = k) = p_{ii}^{k-1}(p_{ii} - 1), \quad k \in \mathbb{N}, \tag{4.2}$$

where p_{ii} is the (assumed constant) probability of staying in the same state. The observed distribution of the run lengths can be seen in Figure 5. This histogram is shown on log-scale, so it is expected to show a straight line. The solid line is the expected distribution based on $p = n_{ii}/n_{i\cdot}$, i.e. the observed number of jumps from state i to state i divided by total number of jumps from state i. It is seen that the overall picture is reasonable, but for both Wake and Delta Sleep there seems to be too many long runs compared to the linear decline in the first part of the histogram.

5 Discussion

We have presented a method for analyzing time series data with a discrete sample space based on the assumption data is generated by a continuous time inhomogeneous Markov jump process. A first objection to this choice of model



(a) Wake (b) Paradoxical (c) Delta Sleep

Figure 5: Histogram of run lengths for data between 14 and 22 hours.

could be: why assume a model in continuous time when data is sampled discretely? In particular when data is sampled equidistantly as in the example in Section 4 it could seem appealing to instead assume a discrete Markov process.

The first answer is that if the process is in fact evolving in continuous time is should also be modelled as such. An example is the observation of sleep cycles from the previous section which is most naturally understood as a continuous process. Also e.g. the example of observations of cloud cover mentioned in the introduction is by nature a continuous process. It is only due to the physical limitations that we are forced to consider discrete observations of the processes.

When the model is handled in continuous time it allows us to apply constraints to the parameterization. This is used in the model for the sleep data in (4.1) where jumps between W and PS are restricted. Another example is found in Madsen et al. (1985) where data from observations of cloud cover is analyzed using a continuous time Markov model. On a short time scale it is known that the weather conditions cannot jump from few to many clouds – the process must visit all stages in between seen on a continuous time scale. This allows a parameterization of the $\boldsymbol{Q}(t)$ matrix where only jumps to neighborings states are possible. If the same process were analyzed in discrete time such a constraint could not be applied, since observations of jumps between all states becomes probable. Constraining jumps to neighboring states is only valid in the limit when the time step approaches zero, and this limit is the continuous representation of the model.

The continuous time model also gives a more natural parameterization of the process, where rates or holding times are estimated instead of probabilities. For most people it is more natural to look at the expected holding time for a state instead of the probability for leaving at the next observation time. Of course these two quantities are to some extent equivalent, and the relation in (3.5) could suggest that one could simply find the continuous representation by solving the equation $\boldsymbol{P}_t = \exp(\boldsymbol{Q}(t)\Delta t)$ for $\boldsymbol{Q}(t)$ for a given estimate of $\boldsymbol{P}_t$. However, this equation will not always have a solution due to the imbedding problem for

Markov chains (Bladt and Sørensen, 2005). In general terms the problem will arise in cases where the process moves to fast between states compared to the sampling rate. This is in fact the case in the sleep example data (Section 4) due to the PS state, which has very short holding times. The problem with the imbedding problem for Markov chains are only avoided by doing the estimation directly in continuous time.

In the formulation of the inhomogeneous Markov model it is assumed that the model dynamics are slowly varying. This formulation does not include a process that has discontinuities in model parameters as a function of time. It could be argued that such a discontinuity is present in the sleep data when lights are turned on after 12 hours. The smooth change in the parameters around 12 hours seen in Figure 3 is thus only an effect of averaging data from both sides of this discontinuity. However, if the estimation around a discontinuity is of particular interest it can be accommodated using standard techniques from local estimation. Either it can be chosen to use a smaller bandwidth where a discontinuity is suspected, or one could choose to use a type of boundary kernel, such that data on one side is not used in the estimation on the other side.

To summarize, the proposed estimation method is aimed at analyzing time series data with a discrete sample space by assuming an inhomogeneous Markov model in continuous time. The formulation of the model in continuous time can reduce the number of parameters in the model by introducing constraints on jumps that exist in the physical system. The estimation method makes it possible to make local non-parametric estimates of changes in model parameters as a function of time. Such changes in the model dynamics can otherwise easily be overlooked or ignored by assuming a homogeneous Markov model, but as illustrated with the sleep data example it can provide valuable information about the underlying process.

References

Anderson, N., S. Garson, S. Fox, S. Doran, B. Ebert, and J. Renger (2005, November). Investigation of effect of route of administration on gaboxadol induced changes in arousal state and eeg parameters. H. Lundbeck A/S.

Bladt, M. and M. Sørensen (2005). Statistical inference for discretely observed markov jump processes. *Journal of the Royal Statistical Society Series B 67*(3), pp. 395–410.

Cox, D. R. and H. D. Miller (1965). *The Theory of Stochastic Processes*. Methuen & Co Ltd.

Fan, J. and I. Gijbels (1996). *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*. Chapman & Hall.

Gasser, T., H.-G. Müller, and V. Mammitzsch (1985). Kernels for nonparametric curve estimation. *Journal of the Royal Statistical Society Series B 47*(2), pp. 238–252.

Madsen, H., H. Spliid, and P. Thyregod (1985). Markov models in discrete and continuous time for hourly observations of cloud cover. *Journal of Applied Meteorology 24*(7), pp. 629–639.

McCullagh, P. and J. A. Nelder (1983). *Generalized Linear Models*. Chapmal & Hall.

Müller, H.-G. (1985). Empirical bandwidth choice for nonparametric kernel regression by means of pilot estimators. *Statistics and Decisions, Supplement Issue*(2), pp. 193–206.

Müller, H.-G., U. Stadtmüller, and T. Schmitt (1987). Bandwidth choice and confidence intervals for derivatives of noisy data. *Biometrika 74*(4), pp. 743–9.

O'Sullivan, F., B. S. Yandell, and W. J. Raynor (1986). Automatic smoothing of regression functions in generalized linear models. *Journal of the American Statistical Association 81*(393), pp. 96–103.

Pawitan, Y. (2001). *In All Likelihood: modelling and inference using the likelihood*. Oxford University Press.

Priestley, M. B. (1981). *Spectral Analysis and Time Series*, Volume 1. Academic Press.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rice, J. (1984). Bandwidth choice for nonparametric kernel regression. *The Annals of Statistics* (4), pp. 1215–30.

Sidje, R. B. (1998). EXPOKIT. A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software 24*(1), pp. 130–156.

Wahba, G. and S. Wold (1975). A completely automatic french curve: fitting spline functions by cross-validation. *Communications in Statistics 4*, pp. 1–17.

Appendix E

Paper E

Title:
Flexible estimation of nonlinear mixed models via Laplace's approximation.

Authors:
S. B. Mortensen and R. H. B. Christensen.

Submitted to:
The Journal of Computational and Graphical Statistics (September 2009).

Flexible Estimation of Nonlinear Mixed Models via the Multivariate Laplace Approximation

Stig Bousgaard Mortensen[1,2] &
Rune Haubo Bojesen Christensen[1]

September 20, 2009

[1]Informatics and Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark. [2]E-mail: sbm@imm.dtu.dk

Abstract

The multivariate Laplace approximation to the marginal likelihood is a fast and accurate approach for estimation in nonlinear mixed models. Our aim is to show how this approximation is easily implemented on a case-by-case basis in general programming environments such as R, S-plus or Matlab. The approach is very flexible compared to what is possible in standard statistical software allowing estimation of models with e.g. crossed random effects and arbitrary correlation structures for the residuals. Such models are not easily, if at all possible, fit with standard statistical software or software specially designed for nonlinear mixed models. The approach also allows graphical assessment of successful convergence and can produce profile likelihoods for selected parameters, neither of which is generally possible with standard statistical software.
Keywords: Crossed random effects, orange tree data, R.

1 Introduction

This paper is concerned with estimation of nonlinear mixed models (NLMMs) where the conditional distribution of the response given the random effects as well as the distribution of the random effects are Gaussian. The model can be expressed generally as

$$(\boldsymbol{Y}|\boldsymbol{B}=\boldsymbol{b}) \sim N(\boldsymbol{f}(\boldsymbol{\beta},\boldsymbol{b}),\boldsymbol{\Sigma}(\boldsymbol{\lambda})), \quad \boldsymbol{B} \sim N(\boldsymbol{0},\boldsymbol{\Psi}(\boldsymbol{\psi})), \tag{1}$$

where $\boldsymbol{\beta}$ are fixed regression parameters, $\boldsymbol{b}$ is a q-vector of random effects, $\boldsymbol{\lambda}$ and $\boldsymbol{\psi}$ are variance parameters parameterizing the covariance matrices $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ and $\boldsymbol{f}$ is the model function. NLMMs can be viewed as a generalization of the ordinary fixed effects nonlinear model (NLM) (Bates and Watts, 1988) to include random effects, and it can be viewed as a generalization of the linear mixed model (LMM) (Laird and Ware, 1982) to allow the conditional mean to be a nonlinear function of the regression parameters. Despite these conceptually small changes, maximum likelihood estimation in NLMMs has been a fair challenge and still is to some extent. The likelihood function of a NLMM is the marginal density of the response where the random effects are integrated out

$$L(\boldsymbol{\theta}; \boldsymbol{y}) = \int_{\mathbb{R}^q} p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{b}) p(\boldsymbol{b}) \, \mathrm{d}\boldsymbol{b}, \tag{2}$$

where p denotes a normal probability density function and $\boldsymbol{\theta} = [\boldsymbol{\beta}^T, \boldsymbol{\lambda}^T, \boldsymbol{\psi}^T]^T$ denotes the vector of fixed parameters. It is this q-dimensional integral that is difficult to solve in general, because approximations have to be invoked. The likelihood can be reduced to a multiple of integrals of lower order when the random parameters arise from only one level of grouping (indexed by i, say); the model can be written as

$$(\boldsymbol{Y}_i|\boldsymbol{B}_i = \boldsymbol{b}_i) \sim N(\boldsymbol{f}_i(\boldsymbol{\beta}, \boldsymbol{b}_i), \boldsymbol{\Sigma}_i), \quad \boldsymbol{B}_i \sim N(\boldsymbol{0}, \boldsymbol{\Psi}), \quad i = 1, \ldots, M,$$

and the likelihood simplifies to a multiple of q_i-dimensional integrals

$$L(\boldsymbol{\theta}; \boldsymbol{y}) = \prod_{i=1}^{M} \int_{\mathbb{R}^{q_i}} p(\boldsymbol{y}_i|\boldsymbol{\theta}, \boldsymbol{b}_i) p(\boldsymbol{b}_i) \, \mathrm{d}\boldsymbol{b}_i,$$

where q_i is the number of random effects for the ith group. In particular, the likelihood reduces to a multiple of one-dimensional integrals when only a single random effect occurs for each group in the data. We will refer to a set of random effects corresponding to a single grouping variable as a random component.

In this paper we shall be concerned with the multivariate Laplace approximation to solve the full q-dimensional integral (2) and thereby allow for any structure of the random effects: scalar or vector-valued random effects, nested, crossed or partially crossed; linear, as well as nonlinear.

Pinheiro and Bates (1995) and Vonesh (1996) studied the Laplace approximation (Barndorff-Nielsen and Cox, 1989; Tierney and Kadane, 1986) for models with a single level of grouping, and Pinheiro and Bates (2000) also studied it for models with nested random effects. Statistical software packages that fit NLMMs (e.g. `nlme` in R and S-plus (Pinheiro et al., 2008), SAS NLMIXED (SAS Institute Inc., 2004)

and NONMEM (Beal and Sheiner, 2004), hereafter denoted "standard software") are designed for models with a single level of grouping or with nested random effects. We show in this paper how the Laplace approximation can be implemented on a case-by-case basis in around 20 lines of code providing fast convergence to accurate maximum likelihood estimates (MLEs) for the general NLMM.

Convergence of NLMMs can be hard to achieve and software can be fooled to declare convergence at a local optimum rather than the global optimum or simply far from optimum due to correlation among the parameters. It is therefore important that the user is able to assess, preferably by graphical methods, that a global optimum has been reached at convergence and whether several local optima with high likelihood exist. We suggest to use pseudo-likelihood curves to facilitate this assessment because they are simple and fast to compute.

While the standard error is a convenient summary of uncertainty in a parameter estimate, it is not always appropriate for regression parameters in NLMMs due to nonlinearities (Bates and Watts, 1988) and almost never appropriate for the variance parameters. Profile likelihood curves and corresponding confidence intervals are natural alternatives to standard errors when these are inappropriate. Regrettably, we do not find them easy to obtain with standard software and it seems that the user is left with likelihood ratio tests of the parameters as the only appropriate inferential tool. Profile likelihoods and corresponding confidence intervals for the variance parameters are easy to obtain from the estimation scheme that we propose.

To illustrate our approach to estimation of NLMMs, we use the orange tree data set that has been used repeatedly in the literature to illustrate estimation of NLMMs. These data and appropriate models for them will be presented in section 1.1. In Section 2 we outline the Laplace approximation and motivate it as a natural approximation to the marginal likelihood for NLMMs. In Section 3 we describe how estimation of NLMMs via the multivariate Laplace approximation can be achieved in few lines of code, and we illustrate the flexibility of the approach. In Section 4 we discuss profile likelihoods and assessment of convergence, and we illustrate how the accuracy of the Laplace approximation can be assessed post hoc. In Section 5 we compare our approach with standard software, and we end with a discussion and conclusions in Section 6.

We illustrate estimation using the statistical programming environment R (R Development Core Team, 2008) and include R-code in the text for illustration, but the approach can be implemented in any functional programming environment (e.g. Matlab) that provides access to a general optimizer (preferably of quasi-Newton type), finite difference approximations to Jacobians and Hessians of user defined functions and allows basic matrix operations.

The complete R-code to produce all fits and figures can be downloaded from `http://imm.dtu.dk/~sbm/nlmm/`.

1.1 The orange tree data set and appropriate models

To illustrate our approach to estimation of NLMMs, we use a study of the growth of orange trees reported by Draper and Smith (1981, p. 524), see the appendix, where the circumference of five trees is measured at seven time points. This data set has been used by Lindstrom and Bates (1990) to illustrate their algorithm, by Pinheiro and Bates (1995) in their comparison of estimation methods and in Wolfinger (1999) to illustrate the NLMIXED procedure. A logistic growth model is fitted in all cases with a single random component b_i allowing for a tree specific asymptotic circumference

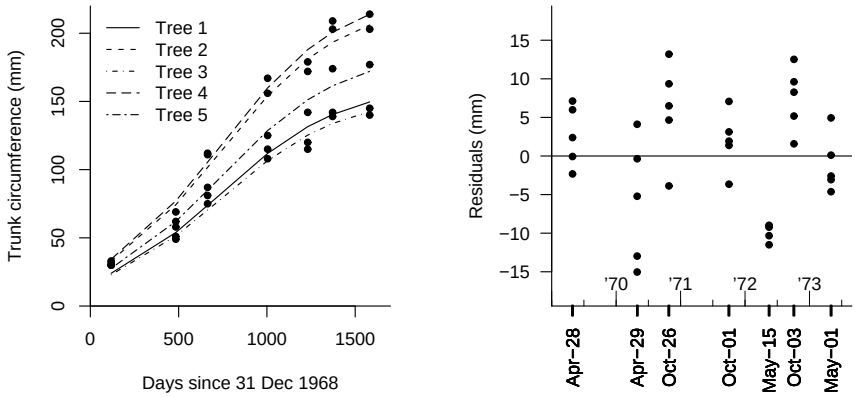
$$y_{ij} = \frac{\beta_1 + b_i}{1 + \exp[-(t_j - \beta_2)/\beta_3]} + \epsilon_{ij} \ , \quad i = 1, ..., 5, \quad j = 1, ..., 7, \tag{3}$$

with $\epsilon_{ij} \sim N(0, \sigma^2)$ and $b_i \sim N(0, \sigma_b^2)$ and mutually independent. The matrices $\mathbf{\Sigma}$ and $\mathbf{\Psi}$ are both diagonal. Here, β_1 determines the asymptotic circumference, β_2 is the age at half this value, β_3 is related to the growth rate and t_j is the time in days since the beginning of the study. The maximum likelihood estimates (MLEs) of the fixed parameters along with standard errors for model (3) are given in Table 1. A plot of the data and model (3) is shown in Figure 1(a).

Table 1: Parameter estimates (standard errors) and log-likelihoods for models estimated in section 3.2 for the orange tree data.

Model	β_1	β_2	β_3	β_4	σ	σ_{b1}	σ_{b2}	ρ	$\log(L)$
(3)	192.1	727.9	348.1		7.84	31.6			-131.57
	(15.7)	(35.3)	(27.1)						
(4)	196.2	748.4	352.9		5.30	32.6	10.5		-125.45
	(19.4)	(62.3)	(33.3)						
(5)	217.1	857.5	436.8	0.322	4.79	36.0			-116.79
	(18.1)	(42.0)	(24.5)	(0.038)					
(4) + (6)	192.4	730.1	348.1		6.12	32.7	12.0	0.773	-118.44
	(19.6)	(63.8)	(34.2)						
(5) + (6)	216.2	859.1	437.8	0.330	5.76	36.7		0.811	-106.18
	(17.6)	(30.5)	(21.6)	(0.022)					

A plot of residuals versus time (sampling occasion) shown in Figure 1(b) reveals an unmodeled variation with time as is also noted by Millar (2004). Millar proposes to include a second random component, b_{2j} for the sampling occasion, that is, crossed



(a) Orange tree data and model fit.

(b) Residuals versus time.

Figure 1: Plots for model (3) for orange tree data.

with the random component for trees. He suggests the model

$$y_{ij} = \frac{\beta_1 + b_{1i} + b_{2j}}{1 + \exp[-(t_j - \beta_2)/\beta_3]} + \epsilon_{ij} \tag{4}$$

with $b_{2j} \sim N(0, \sigma^2_{b2})$ and independent of b_{1i} and ϵ_{ij}, which successfully removes the most significant structure in the residuals. In this model, the effect of the sampling occasion, b_{2j} is proportional to the model prediction. This is reasonable during the initial growth period, but unreasonable when the trees reach their asymptotic circumference. Rather, we find it more natural to include b_{2j} additively in the exp-term in the denominator in model (3) to make the random effects additive on the logit-scale. This makes the effect of the sampling occasion vanish as the trees approach their asymptotic circumference.

A closer look at the sampling scheme reveals, however, that the apparently random effect of the sampling occasion is caused by a seasonal effect and an irregular sampling pattern. In the residual plot in Figure 1(b), it is seen that all samples are taken either in the spring (April or May) or in the fall (October) and that two periods are missing. We include a categorical seasonal effect, β_4 such that

$$y_{ij} = \frac{\beta_1 + b_{1i}}{1 + \exp[-((t_j - \beta_2)/\beta_3 + s_j\beta_4)]} + \epsilon_{ij} \tag{5}$$

where s_j is $-1/2$ and $1/2$ for samples taken in the spring and fall respectively. The models (4) and (5) still show significant unmodeled serial correlation in the residuals within trees. This may be modelled with a continuous auto-regressive (CAR) process

(e.g. Diggle et al., 2002; Pinheiro and Bates, 2000) for the residuals by assuming

$$\text{cov}(\epsilon_{ij}, \epsilon_{ij'}) = \sigma^2 \exp(-\phi|t_{j'} - t_j|/(365/2)), \quad \phi \geq 0 \tag{6}$$

so the full covariance matrix is block diagonal with $\boldsymbol{\Sigma}(\phi, \sigma) = \boldsymbol{I}_5 \otimes \text{cov}(\boldsymbol{\epsilon}_i)$ where $\otimes$ denotes the Kronecker product. The time is scaled so that $\rho = \exp(-\phi)$ can be interpreted as the correlation over half a year and therefore roughly between sampling occasions. Model (4) with crossed random effects cannot easily, if at all, be fitted with standard software for NLMMs. We return to the models described above in Section 3 and show how they can all be easily estimated by means of the Laplace approximation and implemented on a case-by-case basis.

2 The Laplace Approximation

In this section we outline the Laplace approximation to the full q-dimensional integral in the marginal likelihood function (2) for easy reference in later sections. For more details, see Wolfinger and Lin (1997).

The Laplace approximation consists of approximating the logarithm of the integrand in the marginal likelihood (2), i.e. the joint log-likelihood

$$h(\boldsymbol{\theta}, \boldsymbol{b}; \boldsymbol{y}) = \log p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{b}) + \log p(\boldsymbol{b}) \tag{7}$$

by a second-order Taylor expansion,

$$t(\boldsymbol{\theta}, \boldsymbol{b}; \boldsymbol{y}) = h(\boldsymbol{\theta}, \tilde{\boldsymbol{b}}; \boldsymbol{y}) - \frac{1}{2}(\boldsymbol{b} - \tilde{\boldsymbol{b}})^T \boldsymbol{H}(\boldsymbol{\theta}, \tilde{\boldsymbol{b}})(\boldsymbol{b} - \tilde{\boldsymbol{b}})$$

for which the integral has an explicit solution. The expansion is performed around the maximizer of the joint log-likelihood function (conditional mode), i.e. $\tilde{\boldsymbol{b}} = \arg\max_{\boldsymbol{b}} h(\boldsymbol{\theta}, \boldsymbol{b}; \boldsymbol{y})$, which gives the best approximation of the integrand (Barndorff-Nielsen and Cox, 1979). The negative Hessian, $\boldsymbol{H}(\boldsymbol{\theta}, \tilde{\boldsymbol{b}}) = \boldsymbol{f}_b' \boldsymbol{\Sigma}^{-1} \boldsymbol{f}_b'^T - \boldsymbol{f}_{bb}'' \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{f}) + \boldsymbol{\Psi}^{-1}|_{\boldsymbol{b}=\tilde{\boldsymbol{b}}}$, can be approximated by

$$\boldsymbol{D}(\boldsymbol{\theta}, \tilde{\boldsymbol{b}}) = \boldsymbol{f}_b' \boldsymbol{\Sigma}^{-1} \boldsymbol{f}_b'^T + \boldsymbol{\Psi}^{-1}\big|_{\boldsymbol{b}=\tilde{\boldsymbol{b}}}, \tag{8}$$

where the second-order term that usually contributes negligibly (Bates and Watts, 1980) has been omitted. This approximation is the expected Hessian similar to the approximation used in the Gauss-Newton nonlinear least-squares and Fisher scoring methods. The Laplace approximation using $\boldsymbol{D}$ rather than $\boldsymbol{H}$ is refered to as the *modified* Laplace approximation by Pinheiro and Bates (1995) and in its most general

form reads

$$\begin{aligned} l_{LA}(\boldsymbol{\theta};\boldsymbol{y}) &= \log \int_{\mathbb{R}^q} \exp\left\{t(\boldsymbol{\theta},\boldsymbol{b};\boldsymbol{y})\right\} \, \mathrm{d}\boldsymbol{b} \\ &= h(\boldsymbol{\theta},\tilde{\boldsymbol{b}};\boldsymbol{y}) - \frac{1}{2}\log|\boldsymbol{D}(\boldsymbol{\theta},\tilde{\boldsymbol{b}})/(2\pi)|. \end{aligned} \tag{9}$$

There are no constraints on how the random effects enter the model function and thus arbitrary vector-valued, nested, crossed or partially crossed random effects are accommodated.

If the random effects, $\boldsymbol{b}$ appear linearly in the model function, $\boldsymbol{f}$, the Laplace approximation is exact because the second-order Taylor expansion is exact in this case. The modified Laplace approximation (9) is also exact in this case because the second-order term ignored in $\boldsymbol{D}$ is zero. We will sometimes refer to (9) as the *Laplace likelihood* because it depends on the particular model whether it is exact or an approximation.

We can view the Laplace likelihood as an approximation of the integrand in the marginal likelihood (2) by a Gaussian curve. The approximation therefore improves as the integrand, i.e. the joint likelihood, gets closer to a Gaussian curve or equivalently as the joint *log*-likelihood (7) gets closer to quadratic function. Vonesh (1996) shows that as the number of observations increase per random effect, the joint log-likelihood tends to a quadratic function – a fact also supported by the Bayesian central limit theorem (Carlin and Louis, 2000, p.122-124). Because the integral of the joint likelihood is exactly or closely approximated by the integral of an approximating Gaussian curve, we find that the Laplace approximation is a natural approximation for estimation in NLMMs.

3 Model Estimation

This section demonstrates how the Laplace approximation to the marginal likelihood can be used to implement NLMMs on a case-by-case basis with a very limited amount of coding required. This estimation scheme opens up for a great deal of flexibility and enables estimation of a range of models that are not otherwise supported by today's standard software packages. We describe the computational approach in section 3.1 and illustrate the implementation in section 3.2.

3.1 Computational approach

Our computational approach is based on estimating the parameters of the Laplace likelihood (9) by a general purpose quasi-Newton optimizer, for instance of the BFGS-

type (e.g. Nocedal and Wright, 2006). To evaluate the Laplace likelihood (9) for a set of parameters, $\boldsymbol{\theta}$, two quantities $\tilde{\boldsymbol{b}}$ and $\boldsymbol{D}(\boldsymbol{\theta}, \tilde{\boldsymbol{b}})$ has to be available. This leads to a nested optimization, since for every evaluation of the Laplace likelihood with a set of parameters, $\boldsymbol{\theta}$ in the outer optimization, the joint log-likelihood, h has to be optimized over $\boldsymbol{b}$ in the inner optimization. We also use a general purpose quasi-Newton optimizer for the latter task. The only unknown quantity needed to evaluate $\boldsymbol{D}(\boldsymbol{\theta}, \tilde{\boldsymbol{b}})$ given $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{b}}$ is the Jacobian, $\boldsymbol{f}_b'$ for which we use a finite difference approximation. Implementation of any NLMM consists of three functions: The model function, $\boldsymbol{f}$, the joint log-likelihood, h in (7) and the Laplace likelihood in (9).

The starting values in the inner optimization is simply zero; the expectation of the random effects. Starting values for the regression parameters, $\boldsymbol{\beta}$ are based on plots of the data or previous fits of other models, potentially fixed effect versions. Starting values for variance and correlation parameters are qualified guesses based on plots of the data.

At convergence of the outer optimization, we use a finite difference approximation to the Hessian to obtain the variance-covariance matrix of the parameters.

Any inaccuracies in the estimation of $\tilde{\boldsymbol{b}}$ and $\boldsymbol{f}_b'$ are directly reflected as noise in the Laplace likelihood. For the gradient based estimation of the model parameters to converge smoothly, it is therefore important to obtain sufficiently good estimates of these quantities.

The variance parameters are optimized on the scale of the logarithm of the standard deviation to make the estimation unbounded and to make the log-likelihood surface more quadratic facilitating faster convergence. Because all terms in the Laplace likelihood (9) are evaluated on the log-scale, it can be evaluated for any variance parameter arbitrarily close to the boundary at zero (in finite computer arithmetic). This ensures that the optimization will proceed smoothly even if the (MLE) is zero. Further, it allows the likelihood to be profiled with respect to the variance parameters arbitrarily close to zero.

The optimizer `nlminb` in the base package in R is chosen for the inner and outer optimizations. The Jacobian is estimated using the numerical approximation implemented in `jacobian` in the numDeriv package (Gilbert, 2009). The `hessian` function, also from the numDeriv package, is used to obtain a finite difference estimation of the Hessian at the convergence of the outer optimization.

The computational approach described above can be optimized with respect to speed and robustness in a number of respects – generally at the cost of flexibility and more complex coding. Analytical gradients of the joint log-likelihood with respect to the random effects can be found via the general expression $h_b'(\boldsymbol{b}) = \boldsymbol{f}_b'\boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \boldsymbol{f}) - \boldsymbol{\Psi}^{-1}\boldsymbol{b}$ and will increase the speed of convergence of the inner optimization.

The analytical Jacobian will also increase the speed of computation of the Hessian approximation. A Gauss-Newton type estimation of the random effects can replace the quasi-Newton optimization of the inner optimization using the gradient and the Hessian, $\boldsymbol{D}$ from above. This often leads to further speed improvements. In models with many random effects, the quasi-Newton estimation in the inner optimization can benefit from memory-limited BFGS updates (Nocedal and Wright, 2006), and the Gauss-Newton estimation can benefit from the use of sparse matrix methods (e.g. Davis, 2006). Lastly, the residual variance can be profiled out of the likelihood thus reducing the dimension of the outer optimization (Pinheiro and Bates, 1995).

3.2 Implementation in R

In this section we show how the models for the orange tree data presented in section 1.1 can be estimated in a few lines of code with the computational approach for the Laplace approximation described in section 3.1. First, we show how the simple model (3) with a single scalar random effect is implemented. Subsequently we show how this code can be altered in only a few places to fit model (4) with crossed random effects, model (5) with a seasonal effect, and how the CAR process in (6) can be allowed for in the latter models.

The model function, $\boldsymbol{f}$ for model (3) is defined as

```
> f <- function(beta, b) {
      (beta[1] + rep(b[1:5], each = 7))/
      (1 + exp((beta[2] - time)/beta[3])) }
```

The function returns a vector of the same length as the data with model predictions based on the 3 fixed effects in `beta`, the 5 random effects in `b` and the 7 time points in `time`. The joint negative log-likelihood based on (7) is defined as

```
> h <- function(b, beta, sigma, sigma.b) {
      -sum(dnorm(x = circumference, mean = f(beta, b),
                 sd = sigma, log = TRUE)) -
      sum(dnorm(x = b[1:5], sd = sigma.b, log = TRUE)) }
```

using two vectorized calls to the univariate normal density function `dnorm`, because the conditional distribution of the observations and the distribution of the random effects are mutually independent normal. This is the *negative* joint log-likelihood, because standard optimization algorithms by default minimize rather than maximize.

Based on the implementations of the model function and the joint log-likelihood, the Laplace approximation to the marginal log-likelihood $l_{LA}(\boldsymbol{\theta})$ is implemented as

```
> l.LA <- function(theta) {
      beta <- theta[1:3]
      sigma <- exp(theta[4])
```

```
sigma.b <- exp(theta[5])
est <- nlminb(start = rep(0,5), objective = h, beta = beta,
              sigma = sigma, sigma.b = sigma.b)
b <- est$par
h.b <- est$objective
Jac.f <- jacobian(func = f, x = b, beta = beta)
D <- crossprod(Jac.f)/sigma^2 + diag(1/sigma.b^2, 5)
h.b + 1/2 * log(det(D/(2 * pi))) }.
```

where the parameters to be estimated are $\boldsymbol{\theta} = (\boldsymbol{\beta}, \log\sigma, \log\sigma_b)$. The call to `nlminb` in `l.LA` performs the inner optimization and computes $\tilde{\boldsymbol{b}}$, and the Hessian, $\boldsymbol{D}$ is computed as in (8) based on the Jacobian, $\boldsymbol{f}_b'$.

The maximum likelihood fit of model (3) is obtained by performing the outer optimization with the call

```
> fit <- nlminb(theta0, l.LA)
```

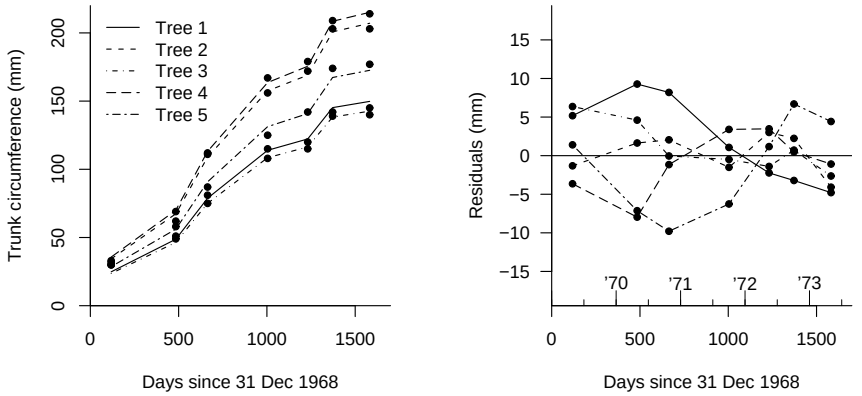
where the starting values, $\boldsymbol{\theta}^0$ are infered from Figure 1(a). The estimation converges in a few seconds on a standard personal computer to the model fit presented in Table 1. This concludes the estimation of model (3). Next, we show how this code can be changed to estimate model (4) with crossed random effects. We mention all changes to the code apart from updates to the parameter sets passed between functions and similar trivialities. Only small changes to the code are required to estimate a model that is not within reach with standard software for NLMMs. This illustrates the power and flexibility of the proposed estimation scheme.

Model (4) has two crossed random components b_{1i} and b_{2j} for tree and time and the full vector of random effects is thus $\boldsymbol{b} = [b_{11}, ..., b_{15},\ b_{21}, ..., b_{27}]^T$. The model function `f` is modified to include the 7 new random effects for sampling occasion by adding the term `rep(b[6:12], 5)` to `beta[1] + rep(b[1:5], each = 7)`. To accommodate the additional random effects with standard deviation σ_{b2} in the joint log-likelihood, the function `h` is updated by adding `-sum(dnorm(x = b[6:12], sd = sigma.b2, log = TRUE))` to the existing code. The only change to the Laplace likelihood, `l.LA` is in the adaption of the change in the covariance matrix for the random effects, $\boldsymbol{\Psi}$ to the Hessian, $\boldsymbol{D}$ in (8); the term `diag(1/sigma.b^2, 5)` is replaced by `diag(c(rep(1/sigma.b^2, 5), rep(1/sigma.b2^2, 7)))`. Model (4) is estimated similarly to model (3) and the optimization converges in a matter of seconds to the results shown in Table 1.

While model (4) is a significant improvement to model (3), model (5) with a seasonal effect might be more appropriate than model (4) with a random effect of sampling occasion. To fit model (5), the only change to the previously defined functions; `f`, `h` and `l.LA` for model (3) is the addition of the term `beta[4] * season` in

the `exp`-term in `f`. The estimate of model (5) is also shown in Table 1. Because the likelihood of model (5) is considerably higher than that of model (4) at the same expense of parameters and producing almost exactly the same predictions, we prefer model (5).

The model fit for model (5) is shown in Figure 2(a). By comparing this to the fit of model (3) in Figure 1(a), it appears that model (5) seems to capture the variation between sampling occasions. This is also verified by a plot of residuals versus time in Figure 2(b), where the residuals within sample occasions are now centered around zero and smaller than in Figure 1(b). The plots for model (4) are very similar to those in Figure 2 for model (5).



(a) Plot of model fit.

(b) Residuals versus time.

Figure 2: Plots for model (5) for orange tree data.

In Figure 2(b), the residuals for each tree have been connected by lines to illustrate that a positive auto-correlation is present. Only small changes to the estimation scheme are required to accommodate any correlation or covariance structure in the residuals. In the following we will describe how the CAR process in (6) for the within tree residuals can be implemented and included in the estimation of the models (4) and (5). We implement the covariance matrix, $\mathbf{\Sigma}$ in (6) as

```
> Sigma.CAR <- function(phi, sigma) {
      diff <- (time[1:7] - rep(time[1:7], each=7))
      delta.t <- matrix(diff / (365 / 2), nrow = 7, ncol = 7)
      P <- sigma^2 * exp( - phi * abs(delta.t))
      kronecker( diag(5), P) }
```

where `delta.t` is a matrix of time differences and `P` is $\text{cov}(\boldsymbol{\epsilon}_i)$. To accommodate the CAR process in the residuals in models (4) and (5), the model functions remain as previously described and the joint log-likelihood is defined as

```
> h <- function(b, beta, sigma, sigma.b, sigma.b2, phi) {
      Sigma <- Sigma.CAR(phi, sigma)
      resid <- circumference - f(beta, b)
      0.5 * (log(det(2*pi*Sigma)) + crossprod(resid, solve(Sigma,
          resid))) - sum(dnorm(x = b[1:5], sd = sigma.b, log=TRUE)) -
          sum(dnorm(x = b[6:12], sd = sigma.b2, log=TRUE)) }
```

where the notable difference from previously is that the first part of `h` is now written as the logarithm of a multivariate normal density using the full residual covariance matrix `Sigma` (in model (5) the last call to `dnorm` concerning σ_{b2} is excluded). The term $(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})$ in the normal density function is computed using `crossprod(resid, solve(Sigma, resid))`, since this is numerically more stable and more efficient than computing the term directly as defined. The only change to `l.LA` to accommodate (6) is in the computation of the Hessian, $\boldsymbol{D}$, where `crossprod(Jac.f)/sig^2` is changed to `crossprod(Jac.f, solve(Sigma, Jac.f))`. To make the estimation of the correlation parameter, ϕ in the CAR process (6) unbounded, it is optimized on the log-scale. The estimates of models (4) and (5) with the CAR process (6) are shown in Table 1. For both models, the CAR process is a significant improvement with p-values <0.001 based on likelihood ratio tests. For model (5), the correlation over half a year, and therefore roughly between sampling occasions, is $\hat{\rho}=0.81$, which is equivalent to the correlation coefficient in a discrete AR(1) model, where account is taken of missing sampling occasions. This corresponds to the strong auto-correlation seen in Figure 2(b) between successive sampling occasions.

4 Graphical Methods for Inference and Validation

In this section we illustrate a number of graphical methods that are useful for inference and validation of a model fit. First, we discuss how to obtain the profile likelihood and corresponding confidence intervals, next we discuss the use of pseudo likelihoods for assessing convergence, and finally we discuss how the accuracy of the Laplace approximation can be assessed when random effects enter nonlinearly in the model function.

The profile likelihood is an inferential tool in its own right, and it can be used to make likelihood based confidence intervals instead of having to rely on the Wald approximation. For a scalar parameter θ, the profile likelihood is defined as $L(\theta)=\max_{\boldsymbol{\eta}} L(\theta,\boldsymbol{\eta})$, where $\boldsymbol{\eta}$ are nuisance parameters and $\boldsymbol{\theta}=(\theta,\boldsymbol{\eta})$. Our approach contains a single loop over the parameter of interest with repeated optimization with respect to the remaining nuisance parameters. The profile likelihood can be interpolated by a spline (e.g. `spline` in R) to reduce the number of values of θ for which the

likelihood has to be optimized to produce a smooth curve. Figure 3(a) shows the relative profile likelihood for the variance parameter for the random effects of sampling occasion in model (4), and Figure 3(b) shows the relative profile likelihood for the half-year correlation ρ in model (5) with the seasonal effect and the CAR residual structure. The horizontal lines at 0.1465 and 0.03625 define 95% and 99% confidence intervals based on the usual χ^2_1-asymptotics of the likelihood ratio statistic. The profile likelihood confidence bounds can be found by numerically solving for the intersection of the spline function with these threshold (e.g. using `uniroot` in R). The figures show which values of the parameters are supported by the data and which are of negligible likelihood relative to the MLE. The figures also illustrate the effect of the arguably arbitrary choice of confidence level.

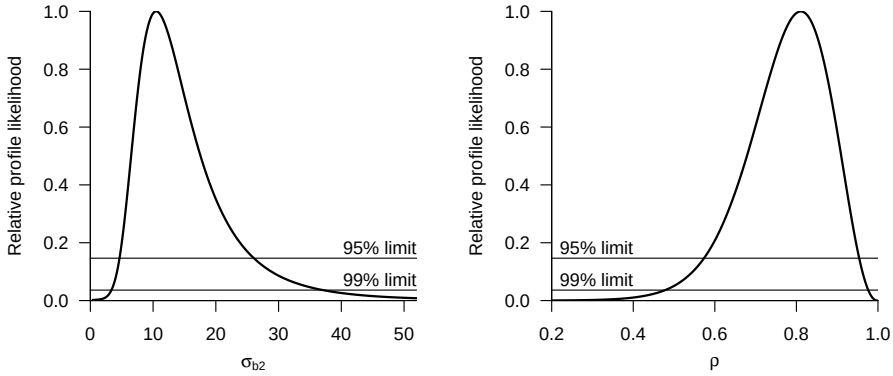


Figure 3: Relative profile likelihoods for (a) the variance parameter for the random effects of sampling occasion in model (4) and (b) the correlation over half a year in model (5) + (6). The horizontal lines indicate 95% and 99% confidence intervals.

The profile likelihood can be time consuming to compute if the parameter set is large because of the many optimizations that are required. If interest is in assessing convergence, the many optimizations can be avoided by making use of the pseudo (or estimated) likelihood. The pseudo likelihood is given as $L_e(\theta) = L(\theta, \hat{\boldsymbol{\eta}})$ where $\hat{\boldsymbol{\eta}}$ is the MLE of $\boldsymbol{\eta}$. It ignores the uncertainty in the remaining parameters, and there is no general way to use it for frequency callibrated inference, but it can be useful for visually checking that the optimization has converged at the optimum. Figure 4 shows pseudo log-likelihood plots for all parameters in model (4) around their MLE on the scale at which the optimization is performed. The plots indicate proper convergence to an unequivocal optimum. If the pseudo log-likelihood is plotted for a very small range around the MLEs, the plots can also be used to study the accuracy in the evaluation of the Laplace log-likelihood as inaccuracies will show up as noise on the

curve. In this way it is found that the error of the log-likelihood surface is on the order 10^{-7} to 10^{-10} for the models we have considered. This is sufficient for the gradient based estimation of model parameters to be both robust and efficient.

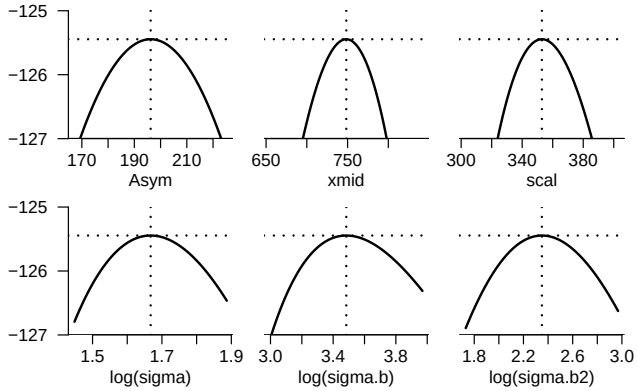


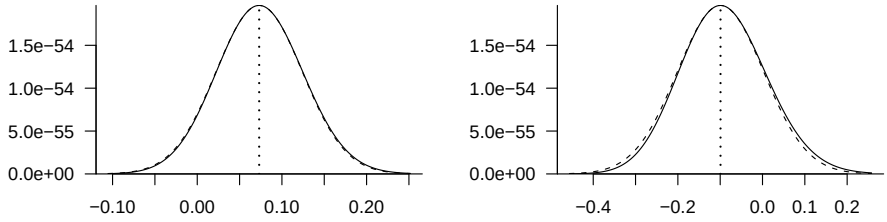
Figure 4: Pseudo log-likelihood profiles for model (4).

The Laplace approximation is exact for all models estimated in Section 3, because the random effects enter linearly in the model functions. Sometimes however, random effects enter nonlinearly and the Laplace likelihood (9) is an approximation to the marginal likelihood (2), and it is of interest to substantiate how accurate the approximation is. We may assess the accuracy of the approximation by graphical means, and to illustrate this, we will use a model that was briefly mentioned in the introduction

$$y_{ij} = \frac{\beta_1 + b_{1i}}{1 + \exp[-((t_j - \beta_2)/\beta_3 + b_{2j})]} + \epsilon_{ij} \tag{10}$$

where b_{2j} enter additively on the logit-scale and thus nonlinearly in the model function. This model has a log-likelihood of -125.39 which is almost identical to that of model (4), cf. Table 1. Because the two random components in (10) are crossed, the integral defining the marginal likelihood (2) is 12-dimensional. We know that the Laplace approximation is exact in the 5 directions corresponding to the linear random effect, b_{1i}. We can examine how good the Laplace approximation to the 12-dimensional integral is in the directions corresponding to the random effects that appear nonlinearly in the model function. This is not a rigid assessment of the accuracy of the entire integral, but intuitively we expect the total error to be small, if the error is negligible in the directions corresponding to the random effects that appear nonlinearly in the model function. Because the random effects in each random component are independent by definition, and because each random effect in one component only depends on the random effects in the other component indirectly

through the fixed parameters, the Hessian, $\boldsymbol{D}$ (and $\boldsymbol{H}$) is very close to orthogonal. Therefore, the error of the Laplace approximation will not be notably larger in the directions that we are not examining. The integrand is given by the joint likelihood, and we may plot this as a function of one of the random effects that enter nonlinearly in the model function, while holding the fixed effects and the remaining random effects fixed at their estimates, $\hat{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{b}}$. The approximating Gaussian curve illustrates the Laplace approximation and is based on $\tilde{\boldsymbol{b}}$ and the appropriate diagonal entry of $\boldsymbol{D}(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{b}})$. Figure 5 illustrates the joint likelihood (solid line) and the approximating Gaussian curve (dashed line) for the random effects b_{24} (a) and b_{27} (b). We expect there to be more information about the random effect b_{24} than about b_{27} due to the structure of the logistic curve (b_{24} is at a sampling occasion, where the slope of the logistic curve is large, and b_{27} is near the asymptotic circumference, where variations has a smaller influence on the model function, cf. Figure 1(a)). This is reflected in Figure 5 in two respects: 1) The curve is wider for b_{27} than for b_{24}, 2) the curve for b_{24} is better approximated by the Gaussian curve than the curve for b_{27}.



(a) b_{24} (b) b_{27}

Figure 5: Joint likelihood (solid line) at the MLE for two random effects that enter nonlinearly in the model function and the Gaussian approximation (dashed line).

Using a general integration function (`integrate` in base package in R), we evaluated the integrals of the joint likelihoods in Figure 5 with sufficient precision and found that the relative error of the Laplace approximation in these directions are 0.08% and 0.22%. The error in the directions corresponding to the remaining random effects that enter non-linearly in the model (10) is of similar size, and we conclude that the error of the Laplace approximation is negligible from a statistical perspective for this model.

In model (10), $\boldsymbol{D}$ is an approximation to $\boldsymbol{H}$ due to the nonlinearity of the model function in the random effects. At the parameter estimates the absolute error in using $\boldsymbol{D}$ rather than $\boldsymbol{H}$ in the term $-\log|\boldsymbol{D}/(2\pi)|/2$ in the Laplace approximation (9) is 0.0058 which is irrelevant from a statistical perspective and lends support to the previous remark that the second-order term ignored in $\boldsymbol{D}$ is of negligible magnitude.

5 Comparison to Standard Software

The Laplace approximation gives the exact marginal likelihood for model (3) for the orange data, because random effects enter linearly in the model function. The model is an example of a simple NLMM with just one random component and can be handled by all standard software. The model is used to compare the accuracy of the R-based estimation scheme to SAS NLMIXED, NONMEM and `nlme` and the results are shown in Table 2. Both SAS NLMIXED and NONMEM were used with the Laplace approximation, and, as can be seen from the table, they both agree with the implementation presented in this paper to all reported digits (NONMEM uses an objective function missing a constant term $\log(2\pi)\sum n_i$ (Wang, 2007), which has been corrected for in the table). Also the parameter and std. err. estimates were found to be very similar. The last row in Table 2 is `nlme` using Lindstrom and Bates' (1990) alternating method, which is not exact for this model. The approximate log-likelihood at the MLE deviates slightly from the others, which is also the case for the parameter estimates.

Table 2: Values of the Log-likelihood for model (3) as reported by various software.

`l.LA` in R	-131.5718851
SAS NLMIXED	-131.57188
NONMEM	-131.5718
`nlme` in nlme	-131.5845

6 Discussion

The presented estimation scheme using the multivariate Laplace approximation offers a very large flexibility in the specification of NLMMs at the cost of only a rather limited amount of coding. It provides an option to fit models, assess convergence and draw inference via profile likelihoods when standard software falls short. Especially models with crossed random effects are not (at least easily) handled by any currently available software package such as NONMEM, SAS NLMIXED, or `nlme` for R/S-Plus. In this way the approach presented here fills a gap left by standard software for NLMMs. The analysis of the orange tree data presented here shows that the flexibility and ability to estimate, compare and draw inference from various models is of substantial importance to the conclusions of the data analysis. The approximation error of the Laplace approximation appears to be of only minor importance.

Several other methods for approximating the marginal likelihood of NLMMs than that of Laplace have been proposed including Gauss Hermite quadrature (GHQ) (e.g.

Davidian and Gallant, 1992), adaptive Gauss Hermite quadrature (AGQ) (e.g. Lui and Pierce, 1994; Rabe-Hesketh et al., 2005), simulation methods and linearization methods. Pinheiro and Bates (1995) compare these methods for models with a single level of grouping and conclude that Laplace and AGQ are the most appealing if one is not content with the linearization method of Lindstrom and Bates (1990). In models with random effects that enter linearly in the model function, the Laplace likelihood is exact, so there is no need to go to lengths with the more computationally demanding AGQ, but AGQ will improve Laplace in models with nonlinear random effects. In models with one level of grouping or nested random components, the integration problems can typically be held in small dimensions, and AGQ can be computationally feasible, but since the number of evaluations of the joint likelihood increases exponentially with the dimension, the scope of the method is limited. A more sophisticated approach is to integrate over a sparse grid rather than the full grid as proposed by Heiss and Winschel (2008). For models with crossed or partially crossed random effect structures, the dimension of the integral increases linearly with the number of random effects, so sparse grid integration also quickly reaches its feasible limit. As an example, consider model (10) with crossed random effects, where one of the random components enter nonlinearly in the model function. The dimension of the integral is 12, and AGQ with a modest 5 quadrature points would require $5^{12} = 244{,}140{,}625$ evaluations of the joint likelihood at each evaluation of the approximation to the marginal likelihood. The sparse grid methods reduce this number substantially so that in 10 and 20 dimensions, the number of points is, respectively, 5,281 and 90,561 (Heiss and Winschel, 2008). For the small orange tree data this would be within range of standard computing power, but if the number of trees is doubled or tripled, this also becomes too inefficient.

Stochastic methods such as simulated likelihood is to some extent applicable to models with crossed or partially crossed random effects, but it also suffers from the curse of dimensionality. As noted by Pinheiro and Bates (1995), the inherent uncertainty in stochastic approximations makes the likelihood hard to optimize. Millar (2004) uses simulated likelihood with 50,000 importance samples and exploiting antithetic variables to estimate model (4). He reports a value of the log-likelihood for model (4) that differ 0.0017 from that of the Laplace likelihood, which is exact for this model with an absolute error less than 10^{-7} as discussed in section 4. However, if the log-likelihood is evaluated at the parameter estimates reported by Millar, the actual error of Millar's simulated likelihood is only 0.0002. Although this difference is irrelevant from an inferential viewpoint, it illustrates the inherent uncertainty in stochastic methods that can hinder optimization of the likelihood.

The estimation times for our implementations are generally longer compared to

those of standard software packages, when these are able to estimate the specified models. The most complex model considered here is model (4) with the CAR structure and it takes a few minutes to fit. The remaining models converges in a matter of a few seconds directly to the MLE without any further effort. Larger data sets will inevitably increase estimation times, but not necessarily the code complexity. In some cases the optimizations of the computational approach mentioned in section 3.1 might be worth the effort.

In this paper we have illustrated the flexibility of estimation with the multivariate Laplace approximation in the framework of NLMMs, but this method is applicable for a much larger class of models. Essentially all mixed models, where the joint likelihood is easily defined can be accommodated including the important generalized linear mixed models; univariate as well as multivariate, and also the less common generalized nonlinear mixed models. When the distribution of the observations is not Gaussian, the Laplace approximation is naturally less accurate, but by the Bayesian central limit theorem (Carlin and Louis, 2000), the joint likelihood tends to a normal curve when the number of observations per random effect increase, so the Laplace approximation can be expected to be fairly accurate when the information per random effect is not small. In our experience the Laplace approximation is remarkably accurate for a wide range of models, but further research is needed to address this topic formally.

Appendix

Table 3: Circumference in millimeters for 5 orange trees reported by Draper and Smith (1981, p. 524).

Tree	Time (days since 31 Dec. 1968)						
	118^S	484^S	664^A	1004^A	1231^S	1372^A	1582^S
1	30	58	87	115	120	142	145
2	33	69	111	156	172	203	203
3	30	51	75	108	115	139	140
4	32	62	112	167	179	209	214
5	30	49	81	125	142	174	177

$^{A,\ S}$: Autumn, Spring

References

Barndorff-Nielsen, O. and D. R. Cox (1979). Edgeworth and saddle-point approximations with statistical applications. *Journal of the Royal Statistical Society, Series B 41*, pp.

279–312.

Barndorff-Nielsen, O. E. and D. R. Cox (1989). *Asymptotic Techniques for Use in Statistics*. London: Chapman & Hall.

Bates, D. M. and D. G. Watts (1980). Relative curvature measures of nonlinearity. *Journal of the Royal Statistical Society, B 42*(1), pp. 1–25.

Bates, D. M. and D. G. Watts (1988). *Nonlinear Regression Analysis and Its Applications*. Wiley, New York.

Beal, S. L. and L. B. Sheiner (2004). *NONMEM® Users Guide*. NONMEM Project Group, University of California, San Francisco.

Carlin, B. P. and T. A. Louis (2000). *Bayes and Empirical Bayes Methods for Data Analysis* (2nd ed.). Chapman & Hall/CRC.

Davidian, M. and A. R. Gallant (1992). Smooth nonparametric maximum likelihood estimation for population pharmacokinetics, with application to quinidine. *Journal of Pharmacokinetics and Biopharmaceutics 20*, pp. 529–556.

Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems*. SIAM, Philadelphia.

Diggle, P. J., P. Heagerty, K.-Y. Liang, and S. L. Zeger (2002). *Analysis of Longitudinal Data* (2nd ed.). Oxford university Press.

Draper, N. R. and H. Smith (1981). *Applied Regression Analysis* (2rd ed.). Wiley, New York.

Gilbert, P. (2009). *numDeriv: Accurate Numerical Derivatives*. R package version 2009.2-1.

Heiss, F. and V. Winschel (2008). Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics 144*, pp. 62–80.

Laird, N. M. and J. H. Ware (1982). Random effects models for longitudinal data. *Biometrics 38*, pp. 963–974.

Lindstrom, M. J. and D. M. Bates (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics 46*, pp. 673–687.

Lui, Q. and D. A. Pierce (1994). A note on gauss-hermite quadrature. *Biometrika 81*, pp. 624–629.

Millar, R. B. (2004). Simulated maximum likelihood applied to non-gaussian and nonlinear mixed effects and state-space models. *Australian & New Zealand Journal of Statistics 46*, pp. 543–554.

Nocedal, J. and S. J. Wright (2006). *Numerical optimization* (2nd ed.). Springer.

Pinheiro, J., D. Bates, S. DebRoy, D. Sarkar, and the R Core team (2008). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-90.

Pinheiro, J. C. and D. M. Bates (1995). Approximations to the nonlinear mixed-effects model. *Jounal of Computational and Graphical Statistics 4*, pp. 12–35.

Pinheiro, J. C. and D. M. Bates (2000). *Mixed-Effects Models in S and S-PLUS*. Springer.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rabe-Hesketh, S., A. Skrondal, and A. Pickles (2005). Maximum likelihood estimation of limited and discrete dependent variable models with nested random effects. *Journal of Econometrics 128*, pp. 301–323.

SAS Institute Inc. (2004). *SAS/Stat 9.1 User's Guide*. Cary, NC: SAS Institute Inc.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association 81*, pp. 82–86.

Vonesh, E. F. (1996). A note on the use of laplace's approximation for nonlinear mixed-effects models. *Biometrika 83*, pp. 447–52.

Wang, Y. (2007). Derivation of various NONMEM estimation methods. *Journal of Pharmacokinetics and Pharmacodynamics 34*, pp. 575–593.

Wolfinger, R. D. (1999). Fitting nonlinear mixed models with the new nlmixed procedure. In *Proceedings of the Twenty-Fourth Annual SAS Users Group International Conference*, Number 287, SAS Institute Inc., Cary, NC. http://ssc.utexas.edu/docs/sashelp/sugi/24/Stats/p287-24.pdf.

Wolfinger, R. D. and X. Lin (1997). Two taylor-series approximation methods for nonlinear mixed models. *Computational Statistics & Data Analysis 25*, pp. 465–490.